SANDIA REPORT
SAND2001-1062
Unlimited Release
Printed April 2001

Final Report for the 10 to 100 Gigabit/Second Networking Laboratory Directed Research and Development Project

Edward L. Witzke, Lyndon G. Pierson, Thomas D. Tarman, L. Byron Dean,
Perry J. Robertson and Philip L. Campbell

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2001-1062
Unlimited Release
Printed April 2001

Final Report for the 10 to 100 Gigabit/Second Networking Laboratory Directed Research and Development Project

Edward L. Witzke, Lyndon G. Pierson, Thomas D. Tarman, L. Byron Dean
Advanced Networking Integration Department

Perry J. Robertson
RF and Opto Microsystems Department

Philip L. Campbell
Networked Systems Survivability & Assurance

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0806

Abstract

The next major performance plateau for high-speed, long-haul networks is at 10 Gbps. Data visualization, high performance network storage, and Massively Parallel Processing (MPP) demand these (and higher) communication rates. MPP-to-MPP distributed processing applications and MPP-to-Network File Store applications already require single conversation communication rates in the range of 10 to 100 Gbps. MPP-to-Visualization Station applications can already utilize communication rates in the 1 to 10 Gbps range.

This LDRD project examined some of the building blocks necessary for developing a 10 to 100 Gbps computer network architecture. These included technology areas such as, OS Bypass, Dense Wavelength Division Multiplexing (DWDM), IP switching and routing, Optical Amplifiers, Inverse Multiplexing of ATM, data encryption, and data compression; standards bodies activities in the ATM Forum and the Optical Internetworking Forum (OIF); and proof-of-principle laboratory prototypes.

This work has not only advanced the body of knowledge in the aforementioned areas, but has generally facilitated the rapid maturation of high-speed networking and communication technology by: 1) participating in the development of pertinent standards, and 2) by promoting informal (and formal) collaboration with industrial developers of high speed communication equipment.

Acknowledgements

The authors wish to thank the following individuals for their contributions to this project: Ron Olsberg (SNL Mission Engineering & Analysis Department), Jean Peña (SNL Advanced Info Architectures Department), Craig Wilcox (formerly of Sandia National Laboratories), Karl Gass (formerly of Utah State University), and Mayfann Ngujo (formerly of New Mexico State University). The authors also wish to thank Mike Vahle and Len Stans for their support of this work.

Contents

Figures

Tables

1 Introduction

The next major performance plateau for high-speed, long-haul networks is at 10 Gbps (OC-192). Data visualization, high performance network storage, and distributed Massively Parallel Processing (MPP) demand these (and higher) communication rates. Previous research has shown that MPP-to-MPP distributed processing applications and MPP-to-Network File Store applications already require single conversation communication rates in the range of 10 to 100 Gbps. MPP-to-Visualization Station applications can already utilize communication rates in the 1 to 10 Gbps range.

Standards are currently evolving, and in some instances, do not exist to meet the needs of distance insensitive, very high bandwidth, secure computer communications. Even though this project helped to evolve some of these standards, there still exist critical technology gaps in the components and systems required to develop and deliver such information in a secure fashion.

This research attempted to facilitate the rapid maturation of technology to fill some of these gaps. Some of these are: 1) Framing Asynchronous Transfer Mode (ATM) cells into Synchronous Optical Network (SONET) Payload Envelopes at high speeds, 2) Segmentation and Re-assembly at different speed regimes, 3) Inverse Multiplexing of ATM and its parallelization features, 4) Assured scalability of ATM/SONET architectures, and 5) Maturation of standards to these ends.

Implementations of communication systems that carry 10 to 100 Gbps exist only as laboratory curiosities. Expensive 10 Gbps (SONET OC-192) optical multiplexers are only now emerging on the market. In order to enable practical communications in the range of 10 to 100 Gbps, the manufacturing cost of 2.5, 10, 40, etc., Gigabit per second electro-optics and Dense Wavelength Division Multiplexing (DWDM) equipment must be greatly reduced; and methods of efficiently processing communication protocol functions in parallel must be developed. As this research progressed, a movement to "streamline" the protocols required to access DWDM channels has emerged. This movement is still proceeding to explore several facets in the marketplace. For example, 10 Gbps ethernet (10GBE), 10 Gbps Packet over SONET (POS), and a more nebulous concept, Packet over Wavelength (POW) are being developed. Some of these technologies will lack "long haul" capability, but will come to play over greater than metropolitan area network distances as optical amplification and "pseudo-regeneration" continue to increase the "transparency diameter" of our optical networks.

2 Architecture

2.1 Architecture Overview, Goals and Scope

Applications such as distributed supercomputing (MPP to MPP), remote visualization (MPP to high performance workstation), and file storage (MPP to network file store) have placed severe bandwidth demands on data communications networks. In fact, the demands on system area networks (SANs), local area networks (LANs), and wide area networks (WANs) exceed the capabilities of today's technologies. This is particularly true for WANs, which attempt to enable distance-insensitive computing.

To address these concerns, Sandia has provided internal R&D funding for the "Ten to One Hundred Gigabit/Second (OC-192+) Network Enabling R&D" project. The purpose of this project is to determine where the technical holes exist which prevent the development of standards and/or products which provide data communications rates in excess of 10 gigabits per second (corresponding to SONET OC-192).

The purpose of this section is to describe an architecture for reliable kernel memory to kernel memory transfers of data between the end systems at rates exceeding 10 Gbps (or SONET OC-192+). Transfers must be *reliable* because applications such as remote visualization, file storage, and MPP simulations require data integrity guarantees. The scope of this project is limited to transfers between the *kernel memories* of each endpoint because variations between implementations of applications and operating systems cannot be completely addressed within the schedule and budget of this project.

2.2 Top-Level OC-192 Architecture

Foremost in the determination of project scope is the definition of a target architecture for OC-192+ communication. Huang describes one such architecture in [31]. Huang's architecture includes transmitters and receivers (i.e., end systems), packet switches, and routers. Expanding on Huang's architecture, Figure 1 shows the architecture that encompasses the work in this project.

Included in this architecture are the endpoints, switches, and layer 3 switch/routers as described in [31], but with additional detail regarding the protocol layer functions that must operate at data rates exceeding OC-192. These layers include the physical links, switching, datagram, and transport layers. A common requirement for each of these layers is that they scale appropriately to meet the increasing bandwidth demands for kernel-to-kernel memory transfers. These layers are described in more detail in the following sections.

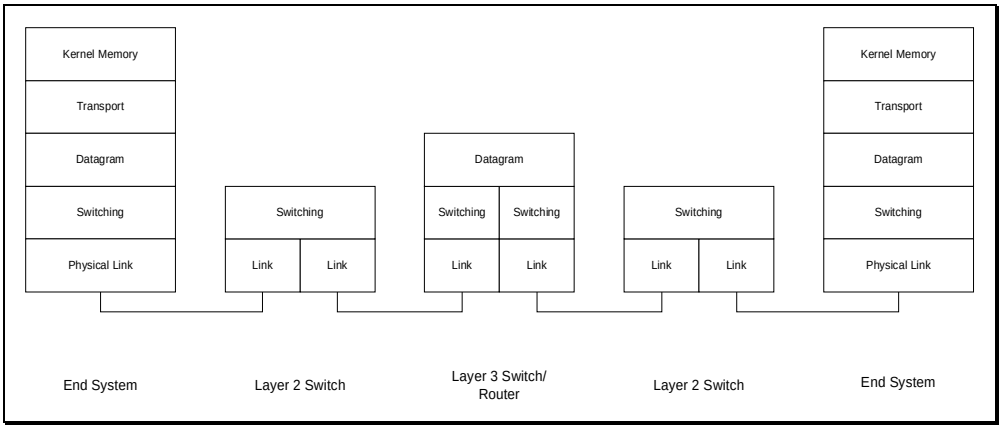


Figure 1. Top-Level OC-192+ Architecture

2.2.1 Physical Links

The physical link hardware is responsible for modulating electrical or optical information onto an optical fiber at rates exceeding 10 Gbps. A number of techniques for scaling physical link hardware to rates above 10 Gbps have been researched, and such hardware is now becoming commercially available.

This is particularly true for Wavelength-Division Multiplexing (WDM), which is now widely used for trunking data and voice circuits over long distances. However, commercially available schemes rely on custom modulation and signaling techniques because standards such as the Switched Optical Network (SONET) have not kept pace with such demands. SONET has successfully scaled from 0.155 Gbps (OC-3) to 10 Gbps (OC-192). Alternatives to SONET are being proposed for rates of 40 Gbps (OC-768 in SONET nomenclature) and beyond.

2.2.2 ATM-to-PHY Interface

With today's ATM physical layer implementations (up to 2.4 Gbps, see [75] and [18]), an interface specification is defined which connects physical layer modules to a circuit board that implements ATM functions (e.g., a network interface adapter or a switch network module). This interface specification is called UTOPIA (Universal Test and Operations Interface for ATM), and allows ATM and/or IP circuit board designers to easily swap SONET I/O modules if the interface rate is to be changed.

At the start of this project, UTOPIA specifications covered bus widths up to 16 bits, which supports SONET link rates up to 0.622 Gbps (OC-12). This project studied the various alternatives in implementing a version of UTOPIA that can support the desired line rates. Comments and contributions were made to the ATM Forum PHY Working Group toward UTOPIA 3, 4, and 5 specifications.

2.2.3 Datagram Adaptation Protocols

Since individual ATM cells provide only 48 bytes of payload, mechanisms are often employed to adapt larger Protocol Data Units (PDUs) to the small cell payload. These mechanisms, called ATM Adaptation Layers (AALs), implement segmentation and reassembly (SAR) to adapt large, variable length PDUs to ATM cells. However, it is not clear whether this process can scale to rates in excess of 10 Gbps due to the overhead associated with reassembly context lookup. This project found few fast hardware implementations of the segmentation and reassembly function. Specialized integrated circuits exist to perform this function at 2.5 Gb/s rates and lower. This project contemplated various mechanisms for implementing AAL SAR functions and required modifications to make AAL functions more scalable. No "breakthrough" innovations for implementing SAR have materialized, however. Other datagram protocols that typically use the services provided by the ATM Adaptation Layer, including IP and the Virtual Interface Architecture (VIA), were also examined.

At the datagram protocol layer, security functions may begin to significantly impact protocol performance. For example, with IP security (IPSEC) and ATM security (ATMSEC), message authentication codes are generated at the source for each packet or AAL5 PDU. Early studies with IPSEC using MD5 message authentication codes indicate serious performance concerns [81].

2.2.4 Security

In addition to the datagram security approaches described above, security functions may be provided at other layers in the 10+ gigabit architecture. This project examined various approaches to security.

2.2.5 Areas of Research

The areas requiring research (technical "holes" or gaps) that have been identified as necessary to enable communications in the 10 to 100 gigabit per second range are:

- Physical layer interfaces
- Physical link hardware
- Inverse Multiplexing into parallel communication links
- Switching and routing
- Expedient inter-process communication
- Network support functions (e.g. encryption, authentication, and data compression)

2.3 OS Bypass and VIA

Traditional mechanisms that implement inter-process communications (IPCs) between separate CPUs require transitions between the application context (also known as "user space") and the operating system context (also known as "kernel space"). This separation has historically provided a number of useful features, including the following:

1. *Abstraction* – the application is not dependent on a specific hardware platform, and upgrades in underlying communications protocols do not require subsequent modification of application programs
2. *Protection* – an application cannot accidentally or maliciously write into another application's memory space
3. *Isolation* – fair scheduling of communication resources (and other resources) by the kernel assures that each application receives its "fair share" of I/O bandwidth, regardless of the behavior of other applications

Since these mechanisms have been traditionally implemented in the operating system, the same CPU that is used to run application programs is also involved in the implementation of the functions listed above. This increased overhead takes CPU cycles away from application programs, causing them to require more time to complete. Although this implementation approach may have made sense in a time when microprocessors were expensive, today's abundance of low-cost, high performance processors and ASICs allows most of these functions to be implemented in the network interface adapter. This shift in communication processing has allowed many researchers to consider the role of the operating system in IPCs in an attempt to provide more CPU cycles to the application, while still preserving the desired attributes of abstraction, protection, and isolation.

Traditional approaches to IPC are also costly because of the transitions between user space and kernel space that were originally regarded as necessary to protect user applications from each other. These transitions often involve much overhead, as process state must be stored on an operating system stack before a system call can proceed with kernel privileges. In addition, when a node receives a message from another node (and process), a significant amount of latency (tens of microseconds) occurs between the time the receive interrupt is generated, and when the application is finally notified of message receipt. Finally, the separation of user processes and the operation system requires that transmitted and received messages must be copied across the user-kernel boundary. For large messages, copying messages across this boundary can lead to significant delay and CPU overhead.

One approach that has been the subject of much research in the past five years is *OS bypass*. As its name suggests, OS bypass can streamline inter-process communications through bypassing much of the associated operating system processing

2.3.1 OS Bypass Background

OS bypass allows an application to perform IPC with an application running on another CPU with very little (if any) involvement by the operating systems running on the transmitting and receiving nodes. Generally, various OS bypass mechanisms strive to achieve the following goals:

1. Low latency – the elapsed time between the time the sending application issues a "send" call to the time the data is available in the receiving application's memory must be minimized.

2. High throughput – the throughput between a single sending process on one CPU to a single receiving process on another CPU must be maximized.
3. Low CPU requirements for communication processing – minimizing communications processing on the host CPU allows for more CPU cycles to be allocated to applications.
4. The traditional goals of protection, isolation, and abstraction – many applications have been written that depend on these functions to be implemented somewhere else, that is, outside of the application.

Of the goals listed above, one of the most important is low latency. As execution rates increase, applications become more sensitive to delays in I/O. These delays can lead to idle time, as receiving applications wait for messages to show up. Most mechanisms described in this report attempt to achieve latencies under 10 μS.

Another important goal is the minimization of communications processing functions performed by the CPU. All of the mechanisms described here attempt to perform this minimization in roughly the same way, by mapping memory regions such that a Network Interface Card (NIC) has direct access to host memory to "stream" data to a receiving NIC/memory region. This approach implements an important CPU overhead-reducing technique: the de-coupling of the CPU from network processing. This de-coupling is important for reducing CPU overhead because network-incurred delays (e.g., scheduling and queuing delays, access delays, speed of light delays, etc.) do not directly cause the CPU to become idle. This is especially important for distance computing applications where large speed-of-light delays exist.

Most of the implementations described here also recognize the fact that interrupts are costly in terms of CPU overhead due to context switching. A widely-adopted approach for minimizing interrupt-related overheads is to perform *interrupt coalescing*. Interrupt coalescing allows one interrupt to be generated, for example, by a NIC when it receives a given number of packets (rather than one interrupt for each received packet). Although this practice reduces CPU overhead, it generally results in higher latency because interrupts are deferred until a specified number of events occur. Therefore, mechanisms that support interrupt coalescing provide this feature as an option that can be configured or disabled by the application and/or the system integrator.

It should be noted, however, that one important goal that has not received much attention in the OS bypass community is standardization. The current lack of standards can be explained by the fact the OS bypass is relatively new, and the current trend of vendor-specific implementations seems adequate at this time. However, this lack of standards can become a serious impediment in the future when heterogeneous clusters become interconnected.

2.3.2 General Solutions

As stated in the Introduction, most of the OS bypass mechanisms described here attempt to minimize CPU communications processing overhead by allowing the NIC to directly

access host memory to transfer memory contents to a receiving NIC/memory region. For OS bypass implementations, there are three important components that must be considered:

- **Applications Programmers' Interfaces (APIs).** APIs provide the interface between the applications and the OS bypass mechanisms.
- **Memory-to-NIC Protocol**. This protocol specifies how a NIC accesses host memory to perform the OS bypass function. This protocol also specifies how the NIC notifies the application when events occur (e.g., the completion of a transmission request, the receipt of data, and the detection of an error).
- **Network Protocol**. The network protocol describes how OS bypass NICs communicate with each other over a network. This protocol may include specifications for data encapsulation, connection establishment/management, routing, and quality of service reservations.

The various OS bypass mechanisms described in this report implement some or all of the above components, depending on the goals of the mechanisms' developers. Table 1 depicts (with a ♦ in shaded table cells) which component is uniquely specified by each of these OS bypass mechanisms. The parenthetical entries in the white (unshaded) table cells indicate "defacto" standards typically used with these protocols. In most cases, the missing components are provided by some other specification (e.g., Fast Messages uses Myrinet to provide NIC and network protocol functions, and ServerNet I and II uses VIA to implement the required API functions).

Table 1. OS Bypass Implementations and Components.

	Operating System Bypass Components		
	API	Memory/NIC Protocol	Network Protocol
Fast Messages	♦	(Myrinet)	(Myrinet)
ServerNet I and II	(VIA)	♦	♦
MINI	♦	♦	(ATM)
Scheduled Transfer			♦
VIA	♦	♦	
Myrinet	♦	♦	♦
SHRIMP	♦	♦	(Paragon Routing Network)

The empty spaces in Table 1 underscore the divergent scope of these OS Bypass approaches and the need for standardization, at least of the interfaces between such modules.

2.3.3 Specific Approaches

2.3.3.1 Fast Messages (FM)

Fast Messages (FM) [53] is an OS bypass mechanism that was developed by Andrew Chien and his colleagues at the University of Illinois. FM is based largely on the Active

Messages (AM) work performed at U.C. Berkeley, and provides a detailed specification for its APIs and functions that closely resembles the AM work. For example, the `fm_send(dest,handler,buf,size)` call allows messages to be transmitted across the network, and associates a handler with the data to be executed before the data is transmitted (to perform additional protocol processing, if required). The ability to associate a handler with data transmission is derived from the AM implementation.

The FM approach provides a number of service guarantees that are normally delegated to the application by other OS bypass mechanisms. These service guarantees are:

- Reliable delivery
- In-order delivery
- Decoupling of the processor and network

FM provides reliability and in-order delivery because they are normally required by applications anyway, and providing these guarantees at the application level results in too much CPU overhead. FM decouples processor activity from the network to allow the application programmer to have more control over communication scheduling, which leads to more control over cache performance, and hence, overall performance.

Like AM, FM does not specify a memory-to-NIC protocol, nor does it specify a network line protocol. However, the FM API imposes more stringent requirements for the underlying network and NIC, including the implementation of flow control, in-order delivery, and buffering. In [53], performance results are provided for an implementation of FM over Myrinet, using native FM APIs, sockets (TCP/IP), and MPI APIs.

An implementation of FM for the Cray T3D is described in detail in [53]. This implementation provides two flavors of FM – *Push FM* and *Pull FM*. The Push FM implementation provides standard, sender-oriented communications. In contrast, the Pull FM implementation is receiver-oriented, that is, the receiver allocates buffer space and sends a request to the sender to transmit data. This eliminates output contention because data is moved only when the receiver is ready to receive it. Although Pull FM provides inherent flow control, the receiver-to-sender request message results in added transfer delay and network overhead.

2.3.3.2 ServerNet I and II

ServerNet is a SAN product that was developed by Tandem Computers, a division of Compaq, for use in their fault-tolerant server products as a processor and I/O interconnect technology. The primary philosophy of ServerNet is to provide a common communications medium for both processor interconnection and I/O (e.g., to disks, TCP/IP networks, etc.). By combining these functions, the same interconnection medium is used, which reduces cost, and increases system reliability and scalability.

Because ServerNet is a product, it addresses all components of an OS bypass implementation, specifically, APIs, memory-NIC interface, and network protocol.

However, since APIs are implemented using the VIA standard, ServerNet APIs are not discussed further in this section.

ServerNet I and II provide the basic functions that are provided by most bus interconnection, including read, write, and interrupts. Both ServerNet versions provide full-duplex I/O. However, the link rate for ServerNet I is 0.4 Gbps, whereas the Link rate for ServerNet II is 1 Gbps. In addition, ServerNet II improves ServerNet I in that it provides a maximum payload of 512 bytes (instead of 64 bytes in ServerNet I), supports 32 and 64 bit addresses (instead of only 32 bit addresses), and provides direct support for VIA.

ServerNet II uses an inter-node address space of 20 bits, and the intra-node address space is 64 bits or 32 bits (32 bit addresses are used in a mixed ServerNet I and II environment). Although the intra-node address could be a physical address, it is usually a virtual address. If a virtual address is used, the intra-node address is processed using an address validation and translation table to map the virtual address to a physical address and check for permissions.

Typically, when a node has a packet to send, the node must know the destination address at the receiving node. However, ServerNet II also supports a "VIA mode" of operation that removes this requirement. In this mode, the receiver posts a receive descriptor, and when data arrives, it is placed into the memory location that is referenced by the receive descriptor. Therefore, the sending node only needs to provide a virtual interface number instead of a 32 or 64 bit address.

The ServerNet network protocol uses hardware-based acknowledgements for data transfer to provide reliable delivery. Therefore, no software is involved in data transfers, unless an error occurs (in which case, driver software will perform error handling).

Finally, the ServerNet protocol provides elaborate network management via in-band messages to implement a variety of functions, including topology discovery, address assignment, fault isolation, performance monitoring, and routing.

2.3.3.3 Memory-Integrated Network Interface (MINI)

The Memory-Integrated Network Interface (MINI) is a network interface adapter that is integrated directly to the memory bus [45]. This choice provides lower latency than traditional network cards that interface to the I/O, and provides the MINI with access to the entire memory address space.

MINI uses standard ATM for its network interconnect. Hosts' physical address regions are directly mapped to ATM virtual circuit identifiers. With this mapping approach, the MINI has the following limits: 4K VCs for 4 Kbyte pages, 2K VCs for 8 Kbyte pages, and 1K VCs for 16 Kbyte pages. In addition, this implementation choice provides good separation because accesses to a given VCI are restricted to those processes that have permission for the corresponding address range. VHDL simulations of the MINI

hardware shows that it can sustain a payload throughput up to 0.96 Gbit/sec, therefore, ATM physical layer interfaces below OC-24 will restrict the throughput of the MINI.

Small message latency with MINI is very good not only because of the direct connection to host memory, but also because MINI implements a priority scheduling algorithm that gives preference to small messages (under 48 bytes). Therefore, when a small message arrives, it will pre-empt any large transfers that may be occurring, and be given a transmission opportunity in the next cell slot. A VHDL simulation of this architecture shows small message, one-way latency of 1.2 μS (excluding propagation and switching delay).

Flow control is provided by MINI using a simple stop/go algorithm. If the receiver's buffer exceeds some threshold, then a "stop" message is sent to the sender. Likewise, when the receive buffer occupancy falls below some threshold, then a "go" message is sent back to the sender.

2.3.3.4 Scheduled Transfer (ST)

Scheduled Transfer (ST) is a protocol that can implement OS bypass transfers over a variety of networks, including Ethernet (and other IEEE 802.2 networks in general), ATM, HIPPI, and HIPPI-6400 [46]. In addition, ST provides support for striping across multiple links. Although the ST protocol and network adaptations are well-defined, it is currently lagging in the specification of an API (although the ST standard committee is considering the specification of a VIA API).

ST actually specifies two message transfer protocols – a short message protocol and a long message protocol. The short message protocol basically operates in the same fashion as packet-based message transfer protocol, i.e., when the sender has a message to send, the ST NIC transfers the message from the sender's memory, and the receiver places that message into a buffer for access by the receiving application. This protocol is ideal for short messages because the long-message scheduling overhead is not invoked, and is particularly appropriate for wide-area message passing because it doesn't involve the long-message flow control.

The long-message protocol, on the other hand, uses an RTS/CTS flow control protocol to ensure that the receiver has sufficient memory allocated for the transfer to successfully complete. This protocol can be implemented either in the host or in the NIC. However, if it is implemented in the NIC, then the CPU is decoupled from the network, and can perform other processing while the transfer is queued for transmission. Although this protocol involves more overhead for transfer scheduling and handshaking, this overhead is amortized over the large message. However, for high bandwidth-delay links, the RTS/CTS handshake may significantly reduce link utilization, and if flow control is implemented in the host, it will increase the amount of time the CPU spends in an idle condition, waiting for the transfer to commence.

For the long message protocol, ST assumes that the memory regions in the sender and receiver are “virtually contiguous”. Therefore, scatter/gather operations (where message data is obtained from and placed into various locations in the sending and receiving hosts’ memory) are not supported.

In addition to the short and long message transfer protocols, ST also defines a method for performing operations on data. This method uses the “FetchOP” message to perform atomic operations on data such as Fetch and Increment, Fetch and Decrement, and Fetch and Clear. These functions are typically useful in protocol processing and inter-host synchronization.

2.3.3.5 VIA

The Virtual Interface Architecture (VIA) is an OS bypass specification developed by the VIA Consortium and its principal members, Intel, Microsoft, and Compaq [24][17]. The Virtual Interface Architecture derives its name from the fact that a VIA network adapter provides each user-level process on the host with a virtual network interface. Each of these virtual interfaces contains separate queues for send and receive descriptors (which contain pointers to transmit and receive data buffers in the host’s memory). This method of separation removes multiplexing and demultiplexing operations from the data transfer path, which results in improved throughput and reduction in delay.

Data transfer primarily occurs through the use of these send and receive queues. Before a data transfer can be initiated, the sending and receiving processes create memory regions to hold the data and pin this memory so that it cannot be swapped to disk. The send and receive memory is then registered with the VIA network interface card (NIC) to allow direct access by the NIC, and to implement memory protections. Once memory is registered with the NIC, a connection is established, which allows the NICs to associate a virtual interface with memory regions and send/receive queues. At this point, the sending and receiving processes post descriptors on the send and receive queues that point to memory buffers that contain the data to be sent, or empty buffers that can accept received data. When the sending process notifies its virtual interface using a *doorbell* register to indicate that the send descriptor is ready to be processed, the NIC transmits the data through the established connection. The receiving NIC obtains the next receive descriptor, writes the received data to the memory region that is referenced by this descriptor, and notifies the receiving process that the receive descriptor was “completed”.

The data transfer model described above implements the basic VIA *send/receive* model. Because this model uses descriptors that can reference multiple memory regions, the sender and receiver can gather and scatter data, respectively, to non-contiguous memory regions. However, VIA also supports a *remote DMA* (RDMA) capability. This data transfer model allows transfers from/to contiguous memory regions, and does not consume descriptors at the receiving virtual interface. For these reasons, scatter/gather is not supported in the RDMA model.

VIA also supports *blocking* and *non-blocking* I/O at the receiver. Blocking I/O allows the receiving process to request that the virtual interface notify it using an interrupt when a posted receive descriptor is completed (i.e., when data is received, or when an error is detected). Non-blocking I/O, on the other hand, allows the receiver to poll the descriptor to determine when it has been completed. The standard tradeoffs between these two descriptor completion notification approaches apply here. Since blocking I/O does not require the receive process to poll for descriptor completion (also known as *spin-waiting*), it results in less wasted CPU cycles. However, non-blocking I/O typically results in faster notification because interrupt latency is avoided.

VIA and ST differ in four important areas – support for scatter/gather, flow control, *applications programmers interface* (API) specification, and low-level networking specification. As stated above, VIA (in send/receive mode) supports gather and scatter operations with send and receive descriptors, which are chained in the virtual interfaces' work queues, and can reference multiple, non-contiguous memory regions. However, ST is more analogous to VIA's RDMA mode in that it requires memory regions to be contiguous, and does not support references to multiple memory regions.

VIA and ST also differ in its use of flow control. In long-message mode, ST requires an RTS/CTS handshake to determine whether sufficient resources exist in the network and at the receiving host. VIA, on the other hand, does not use such flow control. Therefore, ST long-message transfers incur a round-trip latency, which increases with the distance of the transfer, and the number of switches and/or routers that are in the path. However, although VIA does not incur this overhead, data may be lost due to network congestion and/or insufficient available memory on the receiving end. It should be noted, however, that the underlying network technology that is used by VIA may implement end-to-end and/or link-by-link flow control.

Unlike ST, VIA provides a solid specification of the APIs that are used by the sending and receiving processes [17]. However, although the current ST specification [46] does not provide an API specification, it is likely that ST will adopt a subset of the VIA API specification in a future revision.

On the other hand, ST provides specifications for using ST over various link layers, whereas VIA does not. The current VIA specification assumes that the individual vendors will develop the mapping between the specified VIA operations and the various link layer technologies. For homogeneous, static clusters (e.g., database clusters, small compute clusters, etc.), this assumption is valid. However, for large clusters where individual nodes from other locations may join the cluster with various implementations of VIA, the lack of link layer standards may be a problem.

2.3.3.6 Myrinet

Myrinet is a complete, high-speed System Area Network (SAN) protocol developed by Myricom. Therefore, it represents all aspects of an OS bypass technology (i.e., APIs, memory/NIC protocol, and network protocol). The chief design goal for Myrinet is to

minimize switch costs. This goal is realized by moving functions that require intelligence (e.g., topology discovery, routing, address resolution, fault recovery, etc.) away from the switches, and into the hosts' NICs and drivers.

Since routing functions are implemented by the end systems, Myrinet packets are *source-routed*. That is, a packet's route from source to destination is determined by the source, and switching information for each hop in the route is appended to the header of the packet. As the packet traverses the network, each routing field is stripped-off of the header, the header checksum is re-computed, and the packet is forwarded to the next hop switch.

In order for source-routing to work, each end system must have knowledge of the network topology, and be capable of performing address resolution. For Myrinet, one NIC in the network (selected automatically, or by the network manager) performs topology discovery and sends topology packets to all of the other NICs in the network. However, since Myrinet networks are flat, this mechanism will not scale for large networks with dynamic topology (e.g., due to changing cluster membership, network faults, etc.).

Myrinet SAN and LAN links are currently copper-based (though fiber optic interfaces are being developed), and operate up to 1.28 Gbit/sec. SAN links are used primarily for connecting Myrinet NICs to switches, and can extend no further than 3 meters. In contrast, LAN links are primarily used for interconnecting switches, and can extend up to 25 meters. In both cases, the link interfaces at the switches and NICs implement link-by-link flow control. If a source-routed packet is destined for a port that is currently busy, then a flow-control packet is generated, and propagated upstream toward the sender. Each switch receiver implements a small slack buffer to prevent data loss as the flow control message propagates to the sender.

As stated above, the Myrinet switch is a very simple device. It is implemented using a pipelined crossbar architecture, and its worst-case switching latency is 500 ns (during path formation) for small packets. Current Myrinet switches implement 8 ports, though 16 and 32 port switches are being developed.

Myrinet NICs use the LANai chipset, a custom VLSI chip that is used to implement the following Myrinet software functions in the *Myrinet Control Program* (MCP):

- Data scatter and gather
- Myrinet checksum
- IP checksum (which accelerates IP over Myrinet)
- Address to route resolution
- Packet transmission/reception notification
- Network topology discovery

Data transfer functions are implemented in two modes: *zero copy* mode, and *one-copy* mode. Zero-copy mode allows OS bypass techniques using Myrinet's APIs. One-copy mode (i.e., traditional user space to kernel space copy) allows the use of traditional

"sockets" APIs. In both cases, data is copied from host memory to the NIC memory before transmission, so the use of the "zero-copy" term may be misleading. Also, for both data transfer modes, the MCP is in the data transfer path (to perform source routing, address resolution, and checksum computation), therefore, the implementation of the MCP software has a large impact on throughput. However, since the MCP is freely distributed in source code form, modifications can be made (if desired) to achieve desired performance and functionality.

2.3.3.7 Scalable High-performance Really Inexpensive MuliProcessor (SHRIMP)

The Scalable High-performance Really Inexpensive MultiProcessor (SHRIMP) was developed at Princeton as a method of interfacing Pentium PCs to the Paragon routing network. The SHRIMP interface was designed with the goal of providing high bandwidth at low latencies. Their approach is based on virtual memory-mapped communication, which allows message passing between user processes without copying to the kernel (zero-copy).

Two NICs were developed under the SHRIMP project. The first NIC, SHRIMP-I is based on traditional interface designs, and uses software to perform virtual memory mapping. Before a SHRIMP-I transfer can occur, the sending process must map its memory region (using a system call) to an equal-sized memory region on the receiver. When a SHRIMP-I transfer is initiated via another system call, the sending NIC places the receiving physical memory address in the packet header, and DMAs the data from memory out to the Paragon routing network. Upon reception, the receiving NIC DMAs the data to the memory location specified in the packet header, and the receiving process is notified. (As an option, SHRIMP-I also allows the receiver to specify the location for the received data.)

The SHRIMP-II NIC extends the SHRIMP-I NIC's functions by allowing the use of virtual memory addresses rather than physical memory. This enhancement is implemented using a network interface page table, which is analogous to VIA's translation and protection table. In addition, SHRIMP-II supports two memory update policies: deliberate update, and automatic update. The deliberate update policy is initiated in the "normal" fashion, that is, through deliberate notification from the sending process. The automatic update policy provides the NIC with the ability to snoop transactions on the memory bus, and automatically determine when it needs to send data to the corresponding memory region on the receiver. Although the deliberate update policy provides the highest throughput, the automatic update policy provides the lowest latency. Finally, where the SHRIMP-I NIC required a system call to initiate a transfer, the SHRIMP-II interface (in deliberate mode) uses a memory-mapped register to allow a process to directly notify the SHRIMP-II interface when it needs to send data (without kernel involvement).

As stated in [7], the SHRIMP-I interface requires 117 instructions on a Pentium PC to initiate a transfer, and incurs additional system call overhead (additional pages on the transfer require only 26 instructions per page). However, the SHRIMP-II interface

required only 15 instructions, no system call overhead, and only 8 instructions for each additional page. By way of comparison, the Intel NX/2 requires 483 instructions per transfer, plus system call and interrupt overhead.

2.3.4 Current Directions

Products conforming to the VI Architecture (VIA) standard are readily available from a number of manufacturers, including Intel and Giganet. These products are typically found in small server clusters composed of less than 16 nodes. However, the Message Passing Interface (MPI) is now modified to support VIA, allowing applications on large clusters to use VIA interconnects to perform large-scale computations (e.g., sorting). Our simulation results described in [79] confirm VIA's applicability in large clusters.

Another important development in OS-bypass networking is the development of Infiniband. Infiniband is designed to be a high-speed, scalable follow-on to PCI. However, in contrast to the bus architecture in PCI, Infiniband uses a switched architecture, allowing hosts and peripherals (e.g., disks, network, video, and other CPUs) to connect to each other "outside the box". Infiniband allows inter-device transfers to occur without direct OS involvement.

2.4 Dense Wavelength Division Multiplexing

It has been estimated that voice traffic increases about 8% per year and the total data traffic (corporate frame relay, modem dial-up access, etc.) is increasing about 35% per year. Within that, Internet traffic continues a rapid growth exceeding 100% per year. Fortunately, communication carriers have installed fiber optic cables as the backbone of their networks. Hence, implementing time division multiplexing (TDM), carriers now transmit information at a rate of 2.4 Gb/s on a single fiber, and are deploying equipment quadrupling that rate to 10 Gb/s. There are three basic solutions that carriers can consider to add capacity to their networks: more fiber, increase the baud rate of transmission, or increase the number of frequencies of light (wavelength channels) concurrently traveling through the fibers. Since fiber re-deployment is expensive and time consuming, it is not a preferred option. The high-speed electronics that are required to upgrade the capacity of transmission in networks by increasing the baud rate are also expensive. Increasing the number of frequencies of light concurrently traveling through the fibers, or wavelength division multiplexing (WDM), becomes the preferred solution.

Optical fiber exhibits several "transparency windows", or wavelength regions where light absorption is small and optical signals propagate well. These regions are centered around 850 nm, 1310 nm, and 1550 nm. The 800 nm region is used for short range communications because it is not as transparent as the others. Most low cost optical links operate in this region because relatively low cost "CD quality" lasers and Light Emitting Diodes (LEDs) can be made to emit at this wavelength. The 1300 nm region is used for longer range communications because this window is more transparent and because medium cost lasers can be made to emit in this region. Further, the diameter of the fiber waveguide is small enough with respect to the wavelength to enable propagation of a

single wavefront (mode), suppressing other electromagnetic modes that degrade “crisp” pulse risetimes, as the multiple wavefronts interact while propagating down the fiber. This Single Mode (SM) operation is used at 1300 and 1500 nm while Multi-Mode (MM) operation is used for shorter range communications in the 1300 and 800 nm regions. The 1500 nm region is used for the longer range higher speed communications because it has the lowest attenuation and because dispersion correction can be applied to this window, making for less “pulse spreading”. Erbium doped optical amplifiers also operate in this region, making it the preferred transparency window for long-haul communications.

For over a decade two-wavelength WDM has been existing, combining transmission at 1310 nm (the wavelength where traditional single mode fiber has near-zero dispersion) with transmission near 1530 nm (the wavelength region where silica-based single mode fibers have their lowest attenuation). Unfortunately, this method only doubles the capacity of fiber. Extending WDM concepts to tens or hundreds of channels has resulted in the concept of Dense Wavelength Division Multiplexing (DWDM). DWDM is a method that maximizes the information transfer rate by transmitting many multiple signals through the same fiber attenuation window. The most common type of DWDM uses half of a fiber pair for reception and the other half for transmission. There are systems that use single fiber for bi-directional traffic, but these configurations lose some fiber capacity to guard bands preventing channel mixing. Without DWDM, present fiber optic networks will soon be inadequate in providing the new levels of service required.

2.4.1 DWDM Technical Background

DWDM switches (multiplexers/demultiplexers) use lasers tuned to divide incoming data (video, voice or data device) into individual optical wavelengths (colors). These individual optical wavelengths (lightstreams) are sent across a fiber link between two switches. Each lightstream operates as a high-bandwidth pipe, transmitting data at gigabit speeds. The lightstreams are then combined and sent over a single fiber cable to another DWDM switch, which will demultiplex the combined lightstreams.

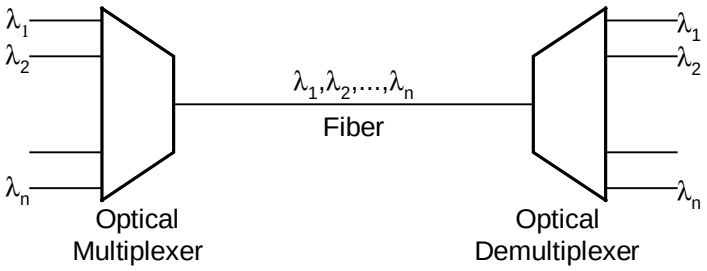


Figure 2. Wavelength Division Multiplexing

Although the concept of DWDM is relatively simple, there are complex combinations of various passive and active components that comprise a DWDM system. Some of the primary passive components are couplers, filters, isolators, attenuators, and fiber. The

passive components create the core of the DWDM system. Consequently, the active components (subsystems) include transmitters, modulators, receivers, add/drop multiplexers, switches, and optical amplifiers.

2.4.2 PASSIVE COMPONENTS

2.4.2.1 Couplers

In a most basic sense, a coupler is an optical device that splits light corresponding to the number of input and output ports. Although there are several types of couplers available, star couplers, broadcast star networks and frequency-selective couplers are discussed in this illustration. Star couplers (N x N) have low excess loss and uniform transmission between the input and output ports and wavelength band of interest. This type of coupler is usually made out of 2 x 2 fiber couplers. Although they have also been made out of planar integrated optics waveguides, this type has a higher loss and needs to be pigtailed. Broadcast star networks can also be created out of N x 1 and 1 x N couplers. Broadcast star networks have an advantage due to their configuration, by allowing growth simply by cascading couplers. Unfortunately, its larger attenuation requires the use of optical amplifiers. Frequency-selective couplers are also another possibility; when the coupling is made frequency-selective, the couplers can be used as a filter or a wavelength router. From a systems point of view, when these couplers are used as routers, they can generally eliminate splitting losses and allow frequency reuse.

2.4.2.2 Filters

Filters are necessary to separate light into many colors. Specifically, fiber Bragg gratings reflect a specific wavelength while transmitting all others. The refractive index of the core in this fiber grating has been permanently altered in a periodic fashion. This is accomplished by exposing it to an optical interference pattern created by an ultraviolet laser. This optical device provides a cost-effective solution to overcome the effects of chromatic dispersion in WDM systems that use erbium doped fiber amplifiers. Fiber Bragg gratings also make it possible for nondispersion shifted fiber to be implemented in the lower attenuation 1550 nm window at data rates of 10Gb/s and greater.

2.4.2.3 Isolators

All optical fibers have impurities that absorb or scatter light. For instance, the hydroxyl ion, which absorbs light at 1400 nm is an impurity. Scattering occurs when the deflection of the transmitted light ray strikes the cladding layer at an angle too large to be reflected back into the core. Isolators are used to eliminate the absorption or scattering that occurs from the optical fibers, and to prevent a wave from propagating in the wrong direction. In effect, it isolates areas of optical sections from one another. Usually isolators allow any light to pass the arrangement only in one direction at one polarization (orientation of the electric field vector of a wave). Fiber optic isolators are designed to provide electrical isolation for a wide variety of electrical signals by fiber optics. Most fiber optic isolator

products are designed for the transmission of digital data, audio, video, and control signals. Isolators are available in both single-stage and dual-stage versions.

2.4.2.4 Attenuators

An intentional reduction of the optical power in a fiber caused by absorption, scattering and geometric disturbance in a fiber is achieved with an attenuator. Attenuators are arranged in the optical path to reduce the signal from 1 to 20 dB, thus allowing detector operation at the optimum power level. Attenuation is used to adjust light levels arriving at a receiver to be within its "dynamic range." The power and light combination in great quantities (too much amplification) can overdrive receivers, which may cause overload situations. Insufficient power and light induces bit errors because receivers are unable to distinguish the real signal.

It is very straightforward why attenuators are used. Usually systems are designed and ordered using worst case specifications and after installation it is frequently discovered that the power received at a particular node might be several dB too high for optimum performance. Since changing the coupler might affect the performance of other nodes, using a low back reflection attenuator can resolve this problem.

2.4.2.5 Fiber

Optical fiber (fiber optic) cables consist of thin filaments of glass (or other transparent materials), that can carry beams of light. These cables can transmit data greater distances without amplification since they are less susceptible to noise and interference than other kinds of cables. However, the glass filaments of the optical fiber are fragile, and require more protection. Therefore, the optical fiber must be run underground rather than on telephone poles.

Optical fiber refers to the medium and the technology associated with the transmission of information as light pulses along a glass or plastic fiber. A laser transmitter encodes electrical signals into pulses of light and sends them down the optical fiber to a receiver, which translates the light signals back into electrical signals.

Optical fiber carries much more information than conventional copper wire. In general, the optical fiber is not subject to electromagnetic interference and the need to retransmit signals except at distance intervals. That is why there are various types of optical fibers that may be utilized. For instance, single mode fiber is used for longer distances while multimode fiber is used for shorter distances. Single mode fiber is a fiber type that supports a single path through its core, thus eliminating other propagation modes that may destructively interfere with each other. In contrast, multimode fiber is a fiber type that supports multiple light paths through its core.

2.4.3 Active Components

2.4.3.1 Transmitters

Transmitters are electronic devices that convert an electrical signal into an optical signal by modulating the current in a light emitting diode (LED) or laser. The LED optical transmitter consists of a forward-biased semiconductor junction diode and a simple transistor transimpedance amplifier circuit. This circuit is called a transimpedance amplifier because it converts a voltage input into a proportional current output. Vertical Cavity Surface Emitting Lasers (VCSELs) are fabricated by multiple layers of GaAs and GaAlAs that form a vertical mirror stack in the semiconductor. An aperture is formed by selective oxidation. The diameter of the aperture is comparable to that of the optical fiber, therefore, the light wave from the semiconductor is coupled with high efficiency into the optical fiber. Usually a lens is used to improve the coupling performance. The light is modulated by modulating the current through the VCSEL. The VCSEL offers several advantages over traditional edge emitting lasers. These devices have lower threshold voltages and higher efficiencies and can be fabricated in mass using traditional integrated circuit fabrication techniques.

2.4.3.2 Modulators and Receivers

Modulators operate by varying a carrier wave to transport an information signal containing voice, data, or image over transmission media in a network. Modulators are implemented as either internal (modulating the source of optical energy) or external (modulating the optical beam after it has been generated). Internal modulators are more compact and less expensive, yet do not operate over extremely high frequencies, due to the relatively large power and voltage swings required to bias the generator into saturation and cut-off. External “Mach-Zehnder” or other modulators are used to modulate a continuous wave source at 10 Gbps and above. As high speed VCSEL technology matures, the need for external modulators is being pushed back to even higher frequencies.

Receivers operate by converting an optical signal to an electrical signal in a specific form, which is then useable by other devices. The optical receiver is a photo-detector and is a reverse-biased semiconductor junction diode. The receiver converts the low-level current output of the PIN photo-detector into a high-level voltage signal. Indium Phosphide and Aluminum Gallium Arsenide photo-detectors achieve higher frequency response than silicon-based photo-detectors.

2.4.3.3 Add/Drop Multiplexers

An add/drop multiplexer (ADM) is capable of extracting or inserting lower rate signals from a higher rate multiplexed signal without completely demultiplexing the signal. By operating the appropriate switches, ADMs are used for coupling one or more wavelength signals from a main input port to one or more drop ports. The other signals are routed to the main output port simultaneously with the signals applied at the add ports of which the

switches are operating. Current ADMs use several discrete electronic and electro-optic components connected with optical fiber.

2.4.3.4 Switches

It is necessary to have switches capable of rerouting light. However, most switches were designed to deal with electrical signals instead of light. Thus, when a beam of light is called from a fiber backbone to a line, the switch has to separate the beam that's being redirected, convert it into electricity, redirect it, convert it back into light, and recombine it into the stream of complex beams. Needless to say this is very costly and time consuming. However, there are a variety of switches that are beginning to emerge capable of working directly with light. Depending on the type of technology, there are switches that are becoming available for a variety of services. For instance, silicon waveguide technology provides an all optical switching alternative that is potentially reliable enough and economical enough to facilitate deployment of local area DWDM.

2.4.3.5 Optical Amplifiers and Regenerators

After a high-powered optical signal travels about 85 km in the "low-loss" 1550 nm transparency window, it has been attenuated to the point that it must be regenerated by some means. This has traditionally been done by optical-electrical-optical converters that discriminate the signal into "ones" and "zeroes" and regenerate the signal, completely re-setting the noise floor for another segment. Therefore, signals that span more than one segment must conform to the format and bit rate supported by the electronic regenerator. The maximum transparency diameter for optical networks (the span over which optical signals can be independent of the regenerator's format) has been limited to less than 85 km. The development of optical amplification has increased the maximum transparency diameter to approximately 400 km.

An optical amplifier's purpose is to amplify the optical signal. The most promising type of amplifier is an erbium doped fiber amplifier (EDFA). EDFAs contain several meters of silica glass fiber that is doped with erbium ions. When the ions are stimulated to a higher energy state, they outnumber those at ground state. This causes the doped fiber to become an active amplifying medium. The erbium ions are pumped to a higher energy state by an infrared pump beam coupled with the signal beam into the erbium. These ions will decay eventually to ground state. Erbium doped optical amplifiers operate efficiently in the 1550 nm attenuation window. Other rare earth dopants such as praseodymium and neodymium show promise for optical amplification in other attenuation windows. Another technique called Raman amplification is also extending the wavelength region (bandwidth) that can be amplified with a single amplifier. [19]

There are two major advantages of Optical Amplifiers (OAs). First, OAs can amplify light without it having to be converted into an electronic pulse. Secondly, an OA amplifies multiple optical channels within its passband, eliminating the need for separate amplifiers for each channel.

After an optical signal has been amplified about five times, the optical noise has been also amplified to the point that it is difficult to discriminate the signal from the noise. This then requires some method of regenerating the information and reducing the noise floor every 5*85 = ~400 km. Advances in all-optical signal re-constitution are now being made. So called "3R Optical Regenerators" [52][88] regenerate (amplify), re-shape, and re-time the optical pulses without actually "discriminating" the signal into "ones" and "zeros". This "nearly digital regeneration" can be applied optically to multiple optical signals within a passband without conversion to electronics, and without need for separate equipment for each channel. In addition, the pulse shapes and pulse timings that can be "re-constituted" through such equipment allow some (but not complete) bit rate and format independence. As the bit rate independence, format independence, signal/noise, and cost improvements for this equipment occur, the transparency diameter (through which arbitrary optical signals can be passed) will increase. Until this transparency diameter increases to be larger than the circumference of the Earth, true digital regenerators that recover and retransmit "ones" and "zeroes" will still be required, imposing some optical signaling structure and standardization on long-haul global communications.

2.4.4 General Classification of Solutions

This project examined many DWDM technologies and products from many DWDM vendors. The technology, markets, and players in this field are moving too fast for timely information to be captured in this report

2.5 IP Switching and Routing

This section studies state-of-the-art IP forwarding technologies, including conventional routing and label switching (a.k.a. IP switching). The objective of this is to determine whether IP will scale to future line speeds of 10-100 Gbps. Several approaches are presented for routing and label switching, and a recommendation will be given as to which approach would best meet the needs of a 10-100 Gbps memory-to-memory data transfer across an internetwork.

2.5.1 IP Background

Internetworking is a method of interconnecting networks with different MAC layer protocols. IP is an internetworking protocol whose functions are used in the 10-100 Gbps architecture to provide connectivity to the global Internet. IP provides a mechanism for forwarding data packets between subnets via a router that takes a packet from one subnet, determines what subnet to which it should be forwarded, then sends the packet to the appropriate subnet with the new subnet's encapsulation. Because of its inherent flexibility and adaptability, IP can support a wide variety of applications on top of many different data link technologies.

In order to provide an answer to whether IP will be able to scale to a 10-100 Gbps architecture, requires analysis of some services or functions that IP provides. One of the major performance costs in the IP suite of protocols is the checksum calculation because it requires reading in each byte of data in a packet and adding it to the sum. There has been some research done to improve header checksum calculations for IPv4. IPv6 does not include header checksums, thereby pushing this problem into higher layer protocols after IPv6 conversion takes place. Another costly IP function is determining a destination path for each packet. A necessary task that a router must perform for path determination is the longest prefix algorithm, and because of the growing size of routing tables, this becomes a very expensive computation. In general, the route lookup process has a worst case cost of $O(\log_k n)$, where n is the number of routes in a table and k is a base indicating the number of routes that can be eliminated by each comparison. Some attempts at improving routing table lookups include caching techniques, algorithms for faster lookups in large tables, and methods for reducing routing table sizes. Other IP functions, such as adding and removing headers, are insignificant in terms of cost compared with those of checksums and route lookups; the worst case cost of these functions are $O(1)$ [54], where l is the length of the header. Since routing table lookups have the highest cost in terms of performance, research in this area that improves the speed of making IP routing decisions will be discussed in this report. The latest routing lookup algorithms will be reviewed as well as approaches to label switching for speeding up or bypassing routing lookups in order to improve routing efficiency.

2.5.1.1 IPv6

Although IPv4 is widely used today, IPv6 is expected to be widely deployed in a few years. There are a number of changes that have been implemented in this next generation of IP. IPv6 packets are designed to simplify packet processing and to help with Quality of Service (QoS) guarantees. The following is a summary of the most notable improvements that have been made to classic IP[22][80]:

Expanded addressing capability – IP addresses have increased from 32 bits to 128 bits in length, creating more addressable nodes. Additional levels of hierarchy are built into the addressing to reduce the number of routing table entries and consequently increase the speed of route lookup. The addressing is designed to simplify auto-configuration and scalability of IP multicast. Also, a new address type ‘anycast’ has been introduced; a packet containing an anycast destination address can go to any one (or more) nodes belonging to the anycast group of IP addresses.

Simplified header format – The IP header format has been simplified and also made to be a fixed length to reduce packet header processing and limit bandwidth cost. The IP header will no longer include a header checksum or header length checking, and fragmentation will no longer be done at the router level.

Improved support for Extensions and Options – IPv6 has a separate Extension Header that makes packet forwarding more efficient, reduces limitations on option length, and allows for more flexibility in adding future options.

Flow labeling capability – IPv6 has a flow label field in its header that allows labeling of flows to identify packet flows that contain data that needs special handling, such as real-time data. By looking up the flow label in a table, a routing device can forward or route a packet without having to examine the entire IP header [55].

Authentication and Privacy capabilities – The addition of the special Extension Headers, Authentication, and Encapsulating Security Payload, are improvements made for preserving authentication, data integrity, and confidentiality.

Although the IPv6 node addresses have more than doubled in size from classic IP addresses, there have been a number of other tradeoffs made in order to improve the efficiency of route table lookups. By incorporating more levels of hierarchy, routing tables should be easier to manage. Also, the addition of the flow label field in the header will provide the option of allowing packets to bypass routing functions in return for faster forwarding for certain flow types, such as real-time video. The IP header is a fixed length, also making it easier to extract information from the header, although route table lookup constitutes for a majority of the routing time. Many of the shortcomings that are inherent in classic IP have been addressed in its next generation.

2.5.1.2 Trends in IP

Current and future applications may demand new network requirements for their function and operation. For example, as more voice and video data is transferred over the network, bandwidth requirements have increased tremendously. IP multicast alleviates some of this demand by reducing network congestion and server loads from data being distributed to multiple destinations. Only one copy of the data has to pass over a network link until the paths diverge, where copies of the data are made and sent on multiple paths to their final destinations.

There are some routing protocols strictly for IP multicast packets, and some of the IP routing protocols have an extension protocol for IP multicasting. Some of these protocols are DVMRP (Distance Vector Multicast Routing Protocol), MOSPF (Multicast Extensions to OSPF), and PIM (Protocol Independent Multicast). The IP multicast routing protocols have to construct a spanning tree for each source and destination group using different methods to find the best route that will take advantage of the benefits that multicasting provides. The performance of IP multicasting depends on how many source/destination-groups there are and how sparse or dense the multicast group is. PIM is under development by an IETF working group, and they have created two versions of PIM, one for densely populated groups and one for sparsely populated groups. Because IP multicasting is not as mature as conventional IP routing, it will take some time to tell how it will be used and how well it scales to high speed.

Another important trend in IP technology is providing performance guarantees which is a requirement of many of today's applications. Some research that has been done in this area includes traffic shaping by the sender to let routers/networks know what kind of traffic to expect, special queuing schemes to allow for bandwidth guarantees, and a setup mechanism to request guaranteed service from the internetwork. These aspects of IP will

not be discussed in this paper except to note that these functions will also have an impact in the scalability of IP to 10-100 Gbps networking.

2.5.1.3 Routing

Routing is an important function of IP that allows networks with different MAC layer protocols to interconnect without having to convert from one protocol to another. The general architecture of a typical router includes the following components, as shown in Figure 3: input ports, output ports, a switching fabric, and a routing processor. The input and output ports receive and transmit packets, respectively, and the switching fabric connects the input and output ports on the various line cards. The routing processor administers the routing protocols and creates and maintains the forwarding table.

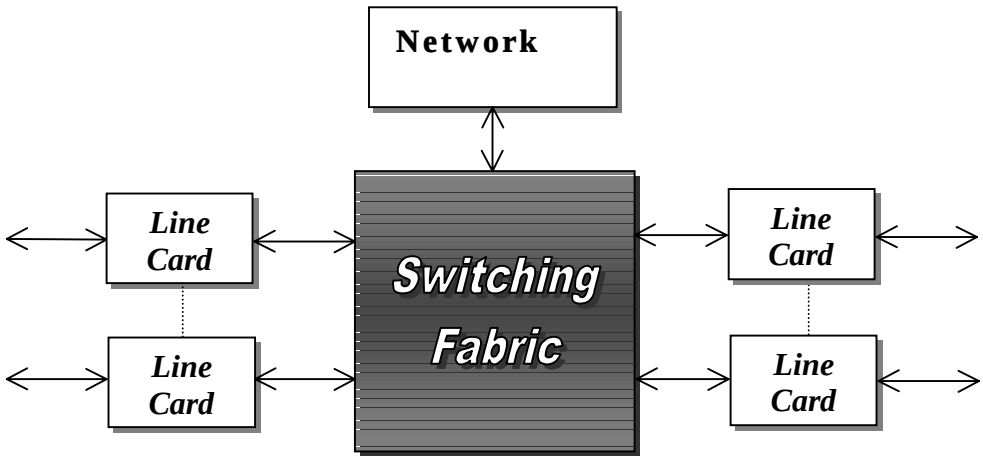


Figure 3. General structure of a high-speed router

Input ports receive incoming packets and perform several functions for the packet. Encapsulation and decapsulation may be needed for the data link layer, depending on whether the packet is going to be forwarded to a network with a different MAC layer. The forwarding engine may be located on the line card itself which gives the capability of looking up the packet's destination port in its own forwarding table. Otherwise, the forwarding engine(s) are located separately, and the line cards share the forwarding engines amongst the group. The input port may also have to classify the packet into a pre-defined service class or run data link layer protocols, e.g. PPP or SLIP. Finally, the input port sends the packet to its destination output port via the switching fabric.

Up until recently, routers used a shared bus, but now gigabit routers have a switched backplane to take advantage of its parallelism [56]. The switching fabric transfers packets from input ports to output ports and can be implemented using one of three common types. A bus connects all input ports to all output ports, but it is limited by its capacity and the overhead associated with sharing a single resource. A crossbar switch provides many data paths between input and output ports; it is essentially 2*N* busses linked with

*N*x*N* crosspoints that are turned on/off for each packet with a scheduler. The limitation of a crossbar switch is the speed of the scheduler. The third commonly used switch type is a shared-memory switch. A shared memory switch stores the packet in memory and switches a pointer to the memory location of the packet. The speed of a shared-memory switch is limited by the speed of a memory access.

The output port stores the packets before they are sent to their destinations. Output ports can implement some kind of scheduling algorithm to support guaranteed service and prioritization of packets. It can also support data link layer encapsulation and decapsulation and higher level protocols.

The routing processor has the task of constructing and maintaining the forwarding tables. It implements routing protocols, e.g. OSPF, BGP, etc., that provide a mechanism for exchanging routing information among neighboring routers and for constructing the forwarding tables. The routing processor also runs software to manage and configure the router [34].

There are three types of packets that a router can potentially forward: unicast packets, unicast packets with Type of Service (ToS), and multicast packets. A unicast packet with ToS is forwarded by using the network layer destination address and applying a longest match algorithm to determine which output to forward the packet to. A unicast packet is forwarded by using a longest match algorithm on the network layer destination address and an exact match algorithm on the ToS value. A multicast packet is forwarded using a longest and exact match on a combination of the network layer source and destination addresses and the incoming interface [49][21].

2.5.1.4 Switching

The creation of label switching was motivated by a search for making faster and cheaper routers. With the growth of the Internet and its continuously increasing number of nodes, routing tables are growing, the link speeds are getting faster, and the routers are having trouble keeping up with the pace. The idea of label switching came from trying to find a way to bypass some of the costly routing functions. It is known that conventional routers are the bottleneck, and it is largely caused by route table lookups performed for each packet that passes through a router. The premise of label switching is to identify a packet flow, to use router functionality to decide where to forward the packet, and to use label swapping techniques much like those of ATM switches to forward all of the subsequent packets of the same flow to the same route. This eliminates the need to do route table lookups for every packet that is essentially going to follow the same route as the first packet. This methodology enables the label switch router to look like a router from a control point of view but perform more like an ATM switch. Label switching also provides some added routing functionality that current routers cannot readily implement, as well as provide a mechanism to integrate IP with ATM [21].

2.5.2 Specific Approaches

2.5.2.1 High speed routing approaches

This section describes four route table lookup algorithms that improve existing router implementations by improving upon the speed of processing IP packets. Unless specified, assume that the approach is using IPv4 addressing.

2.5.2.1.1 DIR-24-8-BASIC

Gupta et al. from Stanford University have come up with a route lookup mechanism that is implemented entirely in hardware and can route a packet in one memory access. Using current 50 nsec DRAM for this implementation, this route lookup mechanism can achieve approximately 20 x 10^6 route lookups per second; using an average packet size of 1000 bits, this routing scheme can support 20 Gbps links [29]. One benefit of this route lookup scheme is that it uses DRAM which is inexpensive and is continually decreasing in price. A second benefit of this scheme is that it is simple and can be implemented entirely in hardware so that the memory accesses can be pipelined. Route prefixes in excess of 24 bits require two memory accesses to lookup, but this algorithm can be implemented in a pipeline fashion so that the effective rate is one memory access per lookup. Also, it has been found by examination of the routing tables of a backbone router that 99.93% of the prefixes in the routing table were 24-bits or less.

This lookup scheme requires two tables that are stored in DRAM; a diagram of the architecture is shown in Figure 4. The first table, TBL24, contains all of the IP address

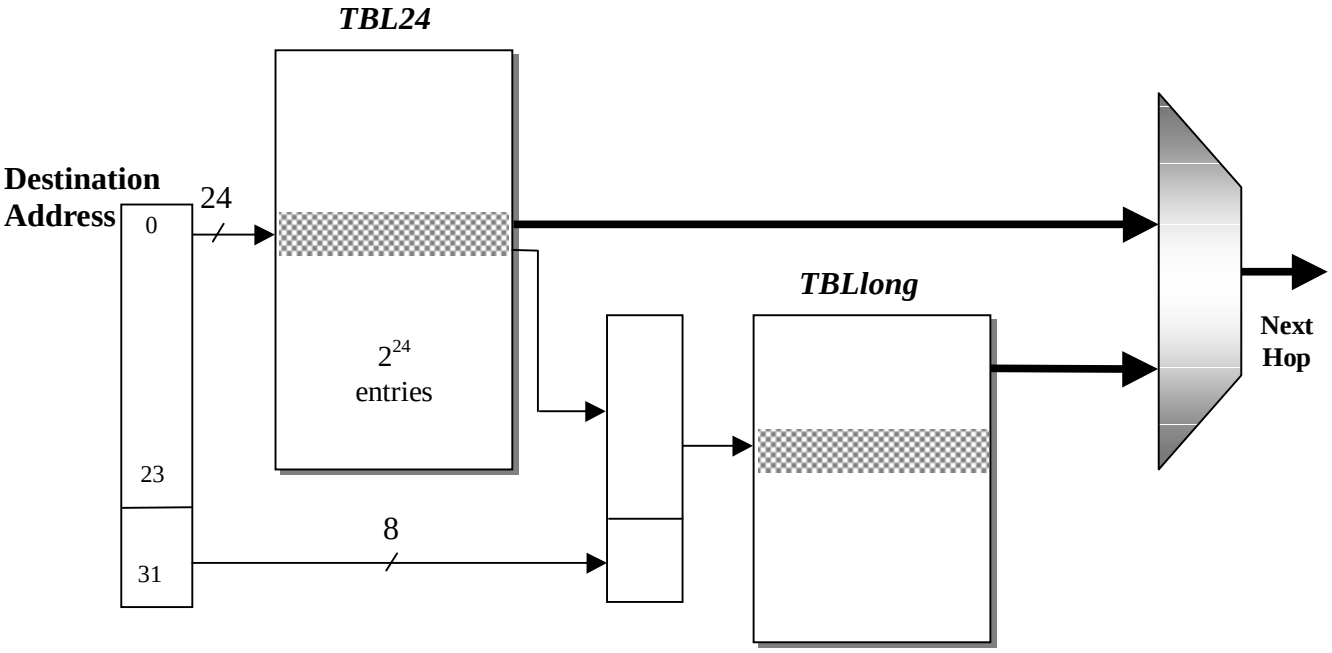


Figure 4. DIR-24-8-BASIC Architecture

prefixes that are 24-bits or less (2^{24} entries). The second table, TBLlong, contains IP address prefixes that are greater than 24-bits

If an address prefix is less than or equal to 24 bits, then the prefix is stored in TBL24, and it has a 16-bit table entry with 0 as the first bit and 15 bits designating the next-hop. If the address prefix is greater than 24 bits, then the first 24 bits of the prefix is put into TBL24, and it has a 16-bit table entry with 1 as the first bit and 15 bits designating a pointer to a set of entries in TBLlong. All prefixes greater than 24 bits have 256 entries allocated in TBLlong. If an IP address corresponds with an address prefix greater than 24 bits, then the TBL24 entry (the last 15 bits) is multiplied by 256, and the product is added to the last 8 bits of the original destination IP address. This value is an index in TBLlong, and the corresponding table entry for this index is the next-hop.

TBLlong uses memory inefficiently because it allocates 256 entries per prefix greater than 24 bits, whether it needs 256 entries or not; for example, a 26-bit prefix would only require 64 entries in TBLlong, but 256 entries are allocated, so 192 entries in the tables will have null values. Although this algorithm uses memory inefficiently, it is still a fast and cheap route lookup mechanism. Variations of this algorithm are presented that use the memory more efficiently by adding additional tables, but the tradeoff is that the complexity of the hardware logic increases and the number of memory accesses increases for each table added. The memory accesses can be pipelined for this case as well.

Gupta et al. also presents different methods for updating the routing tables, three of which were simulated using a sequence of routing updates that were collected from an ISP backbone router. The update schemes were evaluated according to the number of messages that the processor sent out and the number of memory accesses per second that the hardware was required to perform. The results show that the routing table updates impose less than 0.2% of the lookup capacity of the router [29].

2.5.2.1.2 Smaller routing tables

One method of speeding up the route lookup process is to reduce the size of the lookup table. Degermark et al., from the Luleå University of Technology, presents a forwarding data structure that can fit into the cache of a conventional general purpose processor. A routing table with 40,000 entries can be reduced to a forwarding table of 150-160 Kbytes. Since the table is stored in cache, a few million lookups can be performed per second without the need for special hardware. An Alpha 21164 can support a 2 Gbps link using an average packet size of 1000 bits.

The Luleå route lookup algorithm represents a forwarding table in terms of a complete prefix tree that spans all of the entries in the routing table; a complete tree is a tree in which each node has either two children or no children. The tree is divided into three levels (shown in Figure 5): level one includes prefixes of lengths 1-16; level two includes prefixes of lengths 17-24; and level three includes prefixes of lengths 25-32. At the bottom of level one, there are 2^{16} possible nodes. A bit vector of size 2^{16}, or 65536, bits contains information as to whether the prefix is contained within level one or continues to

level two. Using a codeword array, a base index vector, and a map table, one can find a pointer for each possible node that either corresponds to an entry in the next-hop table or to a “prefix chunk”, or additional nodes, in level two of the tree. A similar process is repeated to find a next-hop pointer for a prefix located in level two or level three of the prefix tree.

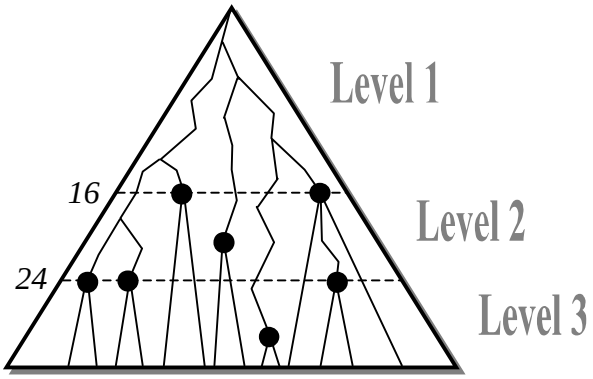


Figure 5. Three levels of data structure.

This routing algorithm has limitations in terms of the size of routing table that this data structure can hold, and the authors point out some modifications that can accommodate more routing entries to an extent. The lookup performance of this algorithm was tested on a 200-MHz Pentium Pro and a 333-MHz Alpha 21164. The algorithm performed 2.0 million and 2.2 million routing lookups per second respectively with the forwarding table in secondary cache for their experimental setup using Internet routing tables available from the Internet Performance Measurement and Analysis (IPMA) project. This algorithm can also be implemented in hardware that would allow pipelining so that two memory accesses could be done in parallel [10]. Another routing lookup algorithm, presented by Waldvogel, et al., performs a binary search via prefix lengths, and it has a projected lookup time (per packet) of 80 nsec for IPv4 and 150-200 nsec for IPv6 packets for their software implementation [86]. This corresponds to 12.5 million route lookups per second for IPv4 addresses and 6.7 million route lookups per second for IPv6 addresses.

2.5.2.1.3 Controlled prefix expansion

Using Controlled Prefix Expansion (CPE) and optimization techniques, the speed of binary searches on prefix lengths can be increased. Srinivasan and Varghese present a route lookup algorithm based on three techniques. Controlled expansion converts a set of prefixes of M different lengths to a set with $N < M$ distinct lengths. In conjunction with CPE, dynamic programming can be implemented to choose optimum expansion levels to reduce storage. Lastly, local restructuring is a collection of techniques that reduce storage of data structures and improve locality. Leaf pushing and cache alignment are local restructuring techniques that, respectively, (1) pre-compute path information and push it

to the leaf, and (2) use packed arrays and semi-perfect hashing to improve binary searches and to limit the amount of collisions while performing hashing functions.

The CPE lookup algorithm was tested against the Luleå scheme using sample routing tables provided by the IPMA project, and it performed faster lookup times (196 nsec vs. 409 nsec) but used more memory (500 KB vs. 160 KB). The advantage that this algorithm has compared with the Luleå scheme is that it, potentially, will be more scalable to IPv6 routing tables because it doesn't have pre-set levels in the prefix trie. This scheme could also be implemented in hardware and pipelined, using 10 nsec SRAM which would be expensive but give better performance [77].

2.5.2.1.4 LC-trie with path and level compression

Another approach to routing lookup methods using trie structures (which are commonly used for parsing text strings or doing longest-prefix route lookups) is called LC-trie with path and level compression, given by Nilsson and Karlsson. Their approach uses the methods of path compression (Patricia trie) and level compression to reduce the size of the prefix trie and speed up next-hop resolution. Path compression is done by removing nodes with only one child and labeling the node with a skip value that indicates how many bits were skipped on that path. This is a method for compressing sparsely populated areas of the trie. Level compression is a technique used to compress parts of the trie that are densely populated. It does so by replacing the i highest complete levels of the trie with a single node with 2^i descendants. The average depth of an LC-trie is proportional to $\log n \cdot \log n$ which means that the LC-trie grows slowly as the number of prefixes grows.

Table 2. Array representation of the LC-trie.

	Node	Branch	Skip	Pointer
→	0	3	0	1
	1	1	0	9
	2	0	2	2
	3	0	0	3
	4	1	0	11
	5	0	0	6
→	6	2	0	13
	7	0	0	12
	8	1	4	17
	9	0	0	0
	10	0	0	1

Each node in the trie is represented with a branching factor, skip value, and a pointer to the leftmost child. As an example, start with string 10110111. Starting at the root (as shown in Table 2), the branching factor is 3, and the skip value is 0, so the first three bits are extracted, 101, and added to the pointer, 1. The result is 6, so the current position is at node 6. At this position, the branch value is 2, so the next 2 bits are extracted with the

value of 2, and 2 is added to the pointer 13. This process is completed until a branch value of 0 is reached which means that the node is a leaf. The pointer value gives the position of the string in the base vector.

This is how the trie structure is traversed, and this method can also be extended to 128-bit addresses for IPv6. Measurements of throughput are given from a Sun Sparc Ultra II which ranges from 1.5 to 2.3 million lookups per second using routing information provided by the IPMA project. This implementation is considered to be a step beyond the previous trie structure given previously by Luleå [51][50].

2.5.2.2 Label Switching

A number of companies came up with their own approaches to label switching, including Cisco, IBM, and Ipsilon. Two of these approaches are described below, as well as the beginnings of a standard for Multiprotocol Label Switching (MPLS) that is currently under construction.

2.5.2.2.1 IP Switching (Ipsilon)

IP Switching is Ipsilon's data-driven approach to the label switching model. Because Ipsilon was one of the first companies to actually implement and market a label switching product, IP Switching is often used as a generic term to refer to label switching. Ipsilon's approach is to leverage the switching speed and QoS capabilities of ATM switches with the functionality of IP routing. The basic idea behind IP Switching is to implement IP on ATM hardware, preserving the connection-oriented nature of ATM and its many advantages over other network layer protocols. The IP Switch identifies and assigns labels to relatively long packet flows that could potentially benefit from IP switching.

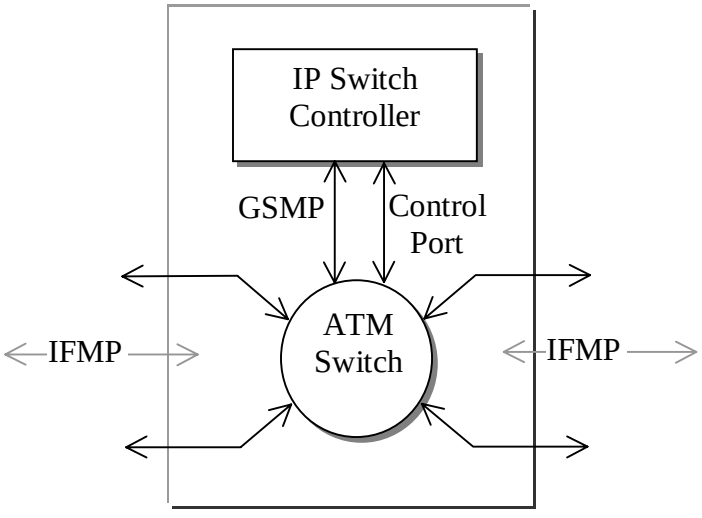


Figure 6. Structure of IP Switch

The IP Switch assigns a Virtual Circuit (VC) to a packet flow, and then subsequent packets belonging to that packet flow are switched to a specified output interface for the given VC, bypassing traditional route lookup algorithm.

An IP Switch has two main components, an IP Switch controller and an ATM switch. The IP Switch controller controls the ATM switch via an open switch management protocol, the General Switch Management Protocol (GSMP). GSMP is a general low-level control protocol to manage ATM switching hardware; GSMP replaces the ATM control software on the switch. Ipsilon partnered with many switch vendors to support their switches with Ipsilon's IP Switch technology; thus, IP Switching can be implemented with a number of commercially available ATM switches using GSMP. The switch controller communicates with the ATM switch via an ATM link using GSMP; the data is encapsulated and sent over the ATM link using AAL5. The main services that the switch controller provides is management of Virtual Circuit (VC) connections across the switch, binding labels to packet flows, and communicating flow binding information to neighboring IP Switches via the Ipsilon Flow Management Protocol (IFMP). The switch controller is a high-end processor that runs IP routing and forwarding software, as well as GSMP, IFMP, and flow classification.

While an IP Switch receives packets, the flow classification and control module observes traffic and tries to identify packet flows that should be label switched. Packet flows that are not selected for label switching are routed normally by the switch controller using standard routing protocols. IP Switches identify two types of packet flows, port-pair flows (Type 1) and host-pair flows (Type 2). Type 2 flows are a sequence of packets that share common IP headers with the same source address and destination address; Type 1 flows are a sequence of packets that share common source and destination IP addresses and source and destination ports. In general, relatively long packet flows benefit most from label switching; short packet flows are not good candidates for label switching because of the initial setup time incurred for the binding process. The policy that the switch controller will adhere to when selecting candidate flows can be tuned by the network administrator, depending on what kind of packet flows should be label switched to improve network traffic flow. The flow classification decision is made independent of other IP Switches.

When a packet flow is identified for label switching, the IP Switch sends an IFMP Redirect message to the upstream IP Switch containing the flow identifier and a VPI/VCI to use for the subsequent IP packets belonging to that flow. The upstream IP Switch will encapsulate the subsequent packets from that flow into AAL5 frames and send them on the redirected VC. IP Switches can also support multicast traffic without modifying any protocols. When a multicast packet is replicated and the packets are sent on their respective paths to their destinations, each of the replicated packets can be redirected by a downstream IP Switch [21][57].

2.5.2.2.2 Tag Switching (Cisco)

Tag switching is another approach to network layer packet forwarding that consequently provides IP scalability. The architecture for tag switching essentially contains two main components: forwarding and control. The forwarding component for tag switching utilizes the simple paradigm of label swapping, and control uses existing network layer protocols plus binding and distribution tag mechanisms. The fundamental interaction of label swapping is between the tag switch and the Tag Information Base (TIB). The tag switch will attempt to match an incoming packet and accompanied tag with one of the entries in the TIB. Each entry consists of an incoming tag plus one or more subentries including outgoing tag, outgoing interface, and outgoing link level information. If the switch makes a match between the a tag in the TIB and the tag carried in the packet, the switch will swap the tag in the packet with the corresponding outgoing tag and outgoing link level information. This information is replaced in the packet, and the packet is forwarded over the associated outgoing interface. There are a few advantages that can be noted for using this simple forwarding paradigm. Since the decision is based on an exact matching algorithm, the search speed is only limited to the length of the tags, which traditionally are fairly short. Therefore, the result is the ability to increase the packet forwarding rate and also to allow for a straightforward approach to hardware implementation. Another advantage to this scheme is that the forwarding decision is independent of the tag's forwarding granularity. This implies that the label-swapping algorithm can not only be applied to unicast but also to multicast packets as well.

A tag may be encapsulated in various ways. The tag header may be inserted in between layer 2 and the network layer headers, in the layer 2 header, or in the network layer header. It is important in tag switching to ensure good binding between a tag and network layer routing; therefore, the main responsibility for the control component of tag switching is to create tag bindings and distribute the tag binding information among the tag switches. The control component is organized as a collection of modules that support particular routing functions. These modules include destination-based routing, multicast routing, and ATM. Tag switching supports destination-based routing by requiring the tag switch to participate in routing protocols (e.g. OSPF, BGP) and creating its Forwarding Information Base (FIB) using the information it acquires from these protocols. Generally, a tag switch will attempt to populate its TIB with incoming and outgoing tags for all routes that are reachable. This implies that tag allocation is accomplished based on topology not traffic, and therefore, it is the mere existence of the FIB entry that determines tag allocation rather than the arrival of data packets.

The fundamental mechanism of multicast routing is the notion of spanning trees. Therefore, for tag switching to support multicast routing, each tag switch must associate a tag with a multicast tree in the following way. When a tag switch creates a multicast forwarding entry and a list of outgoing interfaces for the entry, it also creates a set of local tags, one for each outgoing interface. The tag switch creates an entry in its TIB and places the outgoing tag and interface information plus the locally created tag in the outgoing tag field. This generates a binding between the multicast trees and the tags. The

switch then advertises over all outgoing interfaces associated with the entry and binding between tag and tree.

Since ATM forwarding is based on the label-swapping paradigm, tag switching is regarded as being readily applied to ATM switches by only having to implement the control component of tag switching. An ATM switch can be supported by tag switching, but it requires the implementation of network layer protocols, and the tag switching control component on the ATM switch. To acquire the necessary control information, the switch must be able to participate as a peer in network layer routing protocols [63].

2.5.2.2.3 MPLS

The Internet Engineering Task Force (IETF) has formed a working group whose task is to define a standard approach for label switching called Multiprotocol Label Switching (MPLS). Up until then, numerous proprietary approaches have been taken in designing and implementing label switching which include Tag Switching (Cisco), ARIS (IBM), IP Navigator (Ascend), IP Switching (Ipsilon), and Cell Switching Router (Toshiba). The plan of the work group is to combine the best of the existing approaches and create a label switching standard from that. The following is a description of some of the design capabilities of MPLS that are thus far included in the architecture draft.

MPLS shares a lot of similarities with its predecessors. The basic architecture of a label switch is composed of a control component and a forwarding component, similar to the architecture of a conventional router. The forwarding component's main function is to forward packets from an input port to an output port of a router or switch. It does this by way of two pieces of information: a forwarding table and header information. The forwarding component extracts label *N* from the 'shim header', a header which is placed between the network layer and link layer headers, and accesses the routing information from the *N*th entry of the table. (The placement of the shim header between the IP header and the link layer header effectively places MPLS at a sub-layer between IP and the link layer, as shown in Figure 7.) The label is swapped out for a new label, as well as other

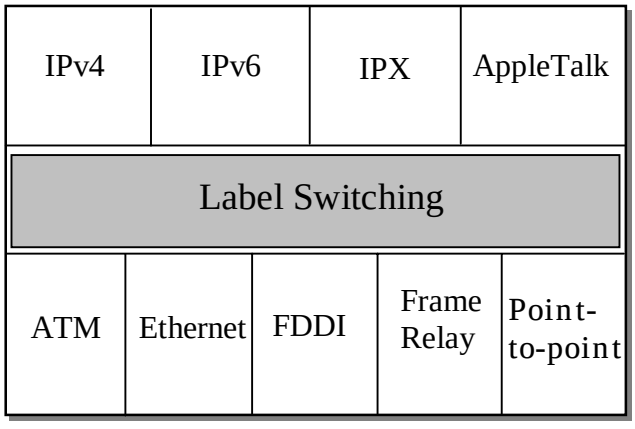


Figure 7. Placement of MPLS

header information, and the packet is sent out to the appropriate interface. The 'shim header' design allows the Label Switched Router (LSR) to be used in conjunction with any link layer and network layer technologies. The LSR can obtain all of the forwarding and resource reservation information with only one memory access.

The control component of a label switch has the tasks of distributing routing information among the LSRs and translating the routing information into a table for the forwarding components to access. An LSR has the features of a conventional router, along with the standard routing protocols, with the addition of software that handles labels, binding, and routing table creation and maintenance. MPLS has control-driven label binding, which means that bindings are created in response to control routing information that is received. The actual binding of a label to a Forwarding Equivalent Class (FEC) occurs at the downstream LSR, and the binding information is communicated to the upstream LSR via the Label Distribution Protocol (LDP). An FEC is a group of IP packets that are to be routed via the same path. The label/FEC pair is only used locally between the upstream and downstream LSRs or in the case of an Label Switched Path (LSP) as described below.

The working architecture document contains some other characteristics of MPLS that will briefly be mentioned. The document defines an LSP as a series of LSRs that link to form a virtual path that share a label/FEC for the entire path which was originally negotiated between the LSP Ingress and the LSP Egress of the path. MPLS allows a packet to contain more than one label, which is referred to as a label stack. One example of how this might be used is if a packet is sent through an 'LSP tunnel' where its route is explicitly defined through a series of LSRs. Each LSR in the path assigns a label to the FEC and the labels are pushed onto the label stack. As the packet travels through each LSR, the LSR pops its label off of the top of the stack (labels are ordered 'Last In First Out') [21][13][1][68][69].

2.5.3 Recommended Approach

Between the two general classifications of IP forwarding (i.e., high-speed routing and label switching), only the literature reviewed for high-speed routing approaches included evidence of performance measures with respect to the speed of the network. Therefore, one solution towards insuring IP scalability for high-speed networks is to choose the approach that claims to have the fastest performance in route table look-ups. The fastest approach was the *DIR-24-8-BASIC* from Gupta et al. at Stanford University. Using an average packet size of 1000 bits to calculate the network speed for each approach, this approach was found to support networks speeds up to approximately 20 Gbps (about 10 times faster than other implementations).

Because the literature reviewed for label switching approaches did not include any figures indicating the supported network speeds, this approach cannot be fairly compared against other IP forwarding solutions. In fact, label switching is a novel idea of trying to bypass the traditional route table look-ups by using packet flows that has tremendous potential to support faster networks. It is therefore imperative that the IP switching

paradigms are tested in a simulation environment and the results compared to the high-speed routing approaches. MPLS appears to be the best solution of the three label switching approaches for various reasons. First, MPLS is the only label switching approach that is not a proprietary solution, which means that commercial products that are based on the MPLS standard will be interoperable. Another factor is that MPLS integrates all of the best label switching paradigms of all of its predecessors, so in principle it should be the best solution.

The answer to the question of whether IP is scalable to 10-100 Gbps network links is yes, to a degree. All of the traditional routing solutions could support gigabit speeds up to 10 Gbps, and two solutions could support speeds exceeding 10 Gbps given an average packet size of 1000 bits. Actually, most of the literature portrays the speed of the device in terms of packets per second (pps), which is more appropriate in this case because even the average packet size varies year to year as traffic patterns and applications change. It is apparent that none of the solutions can currently support more than 20 million pps, and more inventive approaches may be needed to reach higher speeds.

As far as other services that IP provides (i.e. performance guarantees), the literature reviewed does not contain information as to whether these services will scale to 10-100 Gbps link speeds. Most of these schemes are still in the development phase and will most likely be tested in a modeled environment before having the capability of testing these technologies on an operational network. Similarly, because IP multicasting is still at an immature stage, it is not known what the implications will be when it is implemented at high speeds. Also, because it is not yet widely adopted, its projected use and whether it will dominate (percentage-wise) IP traffic, remains to be seen. These factors must be studied further in order to determine whether it will be scalable to high speeds.

With respect to the next generation of IP, it is not currently known what the improved routing tables will look like once IPv6 is in place. Although the addressing scheme has been determined, improvements in the actual size of the route lookup table have not been determined, e.g. whether they will be smaller or not. Also, the transition period between the conversion of IPv4 to IPv6 should be evaluated, since it is not known when and if IPv4 will go away completely. Further study and simulation of IPv6 is needed to determine its scalability to gigabit speeds.

The final recommendations would be to model and simulate the routing approach given by the *DIR-24-8-BASIC* scheme to see how well it can support 10 Gbps links and to what extent it can scale. Also, model and simulate MPLS to determine whether it will support 10 Gbps links and how many VCs it can support given the same environment in which the conventional routing approach is tested. In regard to IP multicasting, this technology should also be modeled and simulated in a similar environment.

2.6 Optical Amplifiers/Transparency Diameter

The telecommunications industry is undergoing revolutionary growth, in terms of both bandwidth capacity and number of mid-level carriers. The exponential growth of the Internet and the emergence of new high-performance optical technologies have enabled this growth. The sustained exponential demand for higher-bandwidth Internet services and greater connectivity has allowed the growth of new entrepreneurial firms (Qwest, Level 3, etc.) that specialize in regional and metropolitan fiber optic infrastructure. Even more established long-haul carriers, that had lit more than 80-90% of their fiber optic infrastructure a few years ago, now have much larger capacity, thanks to the development of Dense Wave Division Multiplexing (DWDM) optical technology. Prototype DWDM switches have demonstrated simultaneous transmission of up to 500 wavelengths of light, each modulated at OC-192 rates (10 Gbps), on a single fiber strand (for an aggregate of 5 terabits/second).

Optical amplification extends the "transparency diameter" of an optical network from approximately 85 km to 400 km [40]. Within this "transparency diameter", a WDM data transport system can achieve true "bit rate and format independence" (i.e., the ability to carry both SONET and non-SONET signals such as optical Gigabit Ethernet, etc.). Beyond this limit, the optical signals must be regenerated to reduce the noise from repeated amplification. Traditionally, this regeneration has been done electrically (interpreting the signal as "ones and zeroes" and re-modulating the optical output), but this is expensive and inefficient for multiple wavelengths carried in a single fiber. Optical regeneration can be applied to all wavelengths simultaneously, resulting in greater relative economy. This involves restoration of pulse height, shape, and timing but not full differentiation of "ones and zeroes", still imposing some restrictions on bit rate and format.

Even though Asynchronous Transfer Mode (ATM) technology was originally designed to integrate the transport of voice, video, and data traffic, the advent of optical amplifiers and regenerators are prompting carriers to consider transport of three separately optimized communication subsystems [9]

2.7 Rapid Processing of Lower Layer Functions

2.7.1 Scaling to Higher Link Speeds using "Inverse Multiplexing"

"Inverse Multiplexing" is a method of implementing a high speed link interface by distributing traffic over multiple lower speed links. A well-defined example is specified in "Inverse Multiplexing for ATM (IMA), version 1.1" [32].

Inverse Multiplexing for ATM (IMA), version 1.1 is an ATM Forum specification for joining low speed (1.5 Mbps, called "T1") ATM links to implement a single link of incrementally higher bandwidth, while coping with delay variation of up to 25 milliseconds and failures/adds/drops of individual links. There are other similar standards emerging to aggregate bandwidth into what appears to be a single link to the customer

equipment (for example, IEEE 802.1 and the link aggregation or "trunking" specification for frame LANs). These methods are distinct from the methods known as “load balancing”.

Load balancing or load sharing (typically employed by IP routers) involves distributing packets over multiple links to provide increased bandwidth between routers. Most routers employ “route caching” to determine how packets will be routed across the available links. By this method, each session gets assigned to a particular port, which limits the throughput per session to the maximum bandwidth available on that port. The inverse multiplexing technique aggregates the multiple links into what appears to be a single high speed link to the customer equipment, making the aggregate bandwidth available to a single application.

Inverse multiplexing also provides graceful degradation of link performance upon failure of one or more links, rather than the time-outs and aborts associated with the re-routing of IP traffic over alternate links by load balancing IP routers.

High speed scientific computing and communication applications are typically interested in the highest single-session throughput possible, therefore this section analyzes the application of inverse multiplexing techniques to aggregate the bandwidth of the highest speed links available (currently 2.5 Gb/s OC-48c). If these links are multiplexed over the same optical path via DWDM methods, then the link delay variation should be acceptably small, even though the bit rates are far faster than the T1/E1 rates targeted by the present ATMF IMA specification.

2.7.1.1 Background on IMA 1.1

Multiplexing is a process whereby multiple, comparatively slow (e.g. T1) lines are combined to create a single, faster data channel that is the aggregate of the lower bandwidths, minus a small amount used for multiplexing overhead. Inverse multiplexing involves the opposite theory, spreading a high speed data path over multiple lower speed (and less expensive) data links. The ATM inverse multiplexing technique involves the multiplexing and de-multiplexing of ATM cells in a cyclical fashion among links grouped to form a higher bandwidth logical link, whose rate is approximately the sum of the link rates. This is referred to as an IMA group.

Some of the attributes of the ATMF IMA 1.1 specification include:

- Appears to customer equipment as a single logical pipe;
- IMA specifies a sophisticated multiplexing scheme that checks individual link performance to ensure the necessary timing requirements (If a link fails to meet the proper characteristics, that link is dropped from the multiplex group.);
- Links can even be added or deleted to a group online, without affecting sessions connected across the IMA group.

The ATMF IMA 1.1 specification is intended for applications that take traffic from a relatively high-bandwidth connection, such as a campus ATM backbone running at

155 Mbps, and spread it across multiple T1/E1 WAN circuits. Even though the ATMF IMA specification calls for equipment to cope with up to 25 milliseconds of link delay variation, some commercial implementations of IMA allow up to 70 milliseconds of link delay variation. This protects users from the delay that can occur under real-world conditions, such as T1/E1 route variability and synchronization issues resulting from service by a variety of carriers.

Figure 8 provides a simple illustration of the ATM Inverse Multiplexing technique in one direction. The same technique applies in the opposite direction.

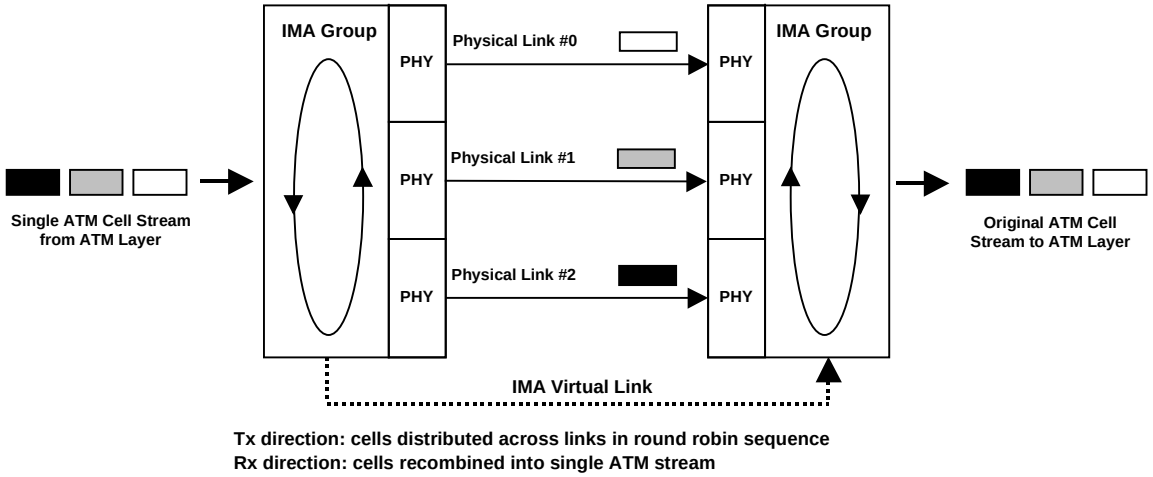


Figure 8. Inverse Multiplexing and De-multiplexing of ATM Cells via IMA Groups

IMA groups terminate at each end of the IMA virtual link. In the transmit direction, the ATM cell stream received from the ATM layer is distributed on a cell by cell basis, across the multiple links within the IMA group. At the far-end, the receiving IMA unit recombines the cells from each link, on a cell by cell basis, recreating the original ATM cell stream. The aggregate cell stream is then passed to the ATM layer.

The IMA interface periodically transmits special cells that contain information that permit reconstruction of the ATM cell stream at the receiving end of the IMA virtual link. The receiver end reconstructs the ATM cell stream after accounting for the link differential delays, smoothing cell delay variation (CDV) introduced by the control cells, etc. These cells, defined as IMA Control Protocol (ICP) cells, provide the definition of an IMA frame. The transmitter must align the transmission of IMA frames on all links as, shown in Figure 9. This allows the receiver to adjust for differential link delays among the constituent physical links. Based on this required behavior, the receiver can detect the differential delays by measuring the arrival times of the IMA frames on each link.

At the transmitting end, the cells are transmitted continuously. If there are no ATM layer cells to be sent between ICP cells within an IMA frame, then the IMA transmitter sends Filler cells to maintain a continuous stream of cells at the physical layer. The insertion of Filler cells provides cell rate de-coupling at the IMA sublayer. The Filler cells should be

discarded by the IMA receiver. A new OAM cell is defined for use by the IMA protocol. This cell has codes that define it as an ICP or Filler cell.

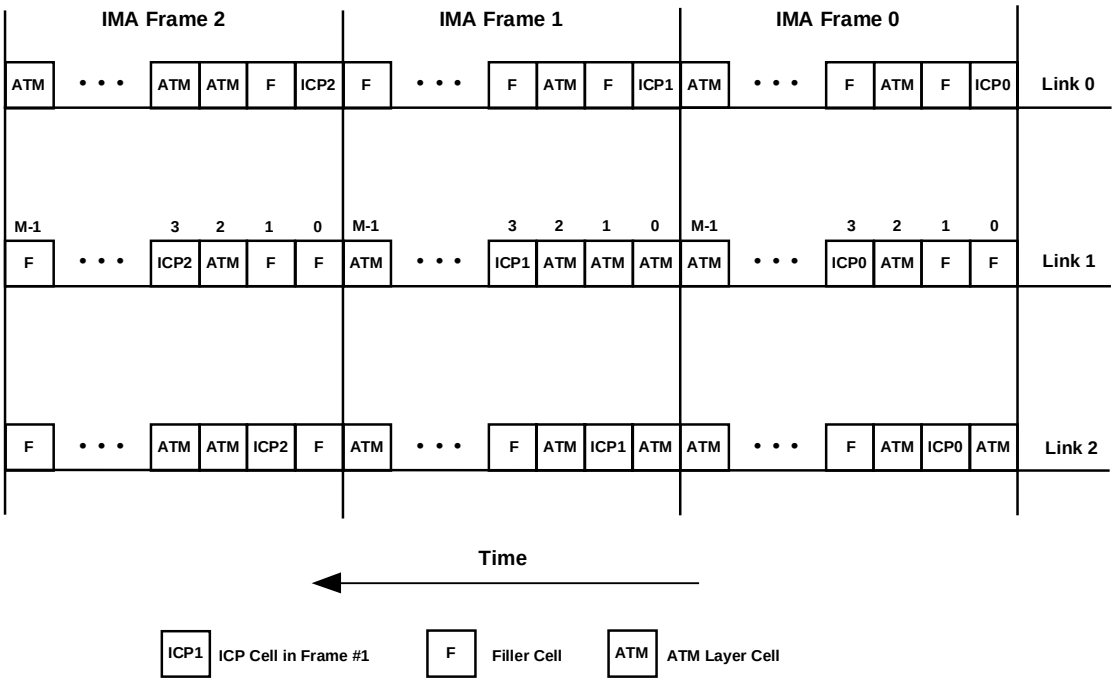


Figure 9. Illustration of IMA Frames

2.7.1.2 IMA Results

Project personnel participated in the refinement of the ATMF IMA standard for aggregating 1.5 Mbps links into larger channels. Separately (from the ATMF effort), these concepts were taken further, creating a "top-level" design for fast reconfigurable hardware to demultiplex four 2.5 Gbps cell streams into a single 10 Gbps cell stream (and vice-versa). A circuit board utilizing UTOPIA 4 (UL4) 10 Gbps interfaces and Low Voltage Differential Signaling (LVDS) I/O pins on Programmable Logic Devices (PLDs) was designed. A survey of the available PLDs with LVDS I/O revealed that advanced components from one manufacturer (Altera) met more stringent I/O specifications but only for 32 of the 34 signal lines required to implement a full 10 Gbps UL4 interface. An alternate manufacturer (Xilinx) marketed components with marginal I/O specifications (for this task), yet provided the potential for multiple sets of the 34 signal lines required per 10 Gbps interface. The prototype design was intended to investigate 1) the feasibility of driving these signals through a connector (such as to a "daughtercard") that would implement specialized I/O technologies such as "Packet over SONET", "ATM over SONET", etc. and 2) the feasibility of IMA multiplexing at 10 Gbps. Wiring schematics and preliminary board layout were developed, but project funding limitations precluded construction of this prototype.

There are two questions not addressed by the above prototype design that may be the subject of future research. First, how large are the buffers required? Second, how can the IMA technique be adapted to IP packets?

How large must the data buffers for each of the multiple paths be in order to compensate for delay differences between the different 2.5 Gbps paths that would be demultiplexed into a single 10 Gbps (or higher speed) channel? If these channels were routed over different physical paths then infeasible large buffers would be required. Latency differences of 25 ms (as specified in the current IMA standard) would require approximately 8 Megabytes of buffer memory per 2.5 Gbps channel (compared to 5 Kilobytes per 1.5 Mbps channel). Latency differences for DWDM channels carried over the same fiber should be extremely small (but not negligible at these data rates). Measurements of channel to channel latency for delivery of ATM Cells over DWDM channels should be made in order to determine this design parameter.

Theoretically, the inverse multiplexing technique can be applied to higher layer protocols (such as IP) carrying large amounts of data. This may be needed in order to eliminate and/or streamline the lower layer protocols for higher efficiency and cost effectiveness. How would one implement a 10 Gbps "Packet over SONET" link out of multiple 2.5 Gbps "Packet over SONET" channels, without resorting to ATM connectivity? As discussed earlier, simply connecting multiple 2.5 Gbps channels to an IP router can result in greater aggregate bandwidth but not in greater bandwidth available to a single session. "Inverse Multiplexing of IP" would present the aggregate bandwidth as a single port to a router, enabling greater single session bandwidth without complex changes to the routing protocols. However, the queuing delay incurred while processing the variable length of each packet will add to the channel-to-channel latency, increasing the size of buffers required. In addition, re-ordering of packets may occur in such an inverse multiplexing scheme for variable length packets. At first this would appear not to be a problem, since IP is not designed to maintain packet ordering, and the end system is responsible for re-assembly of datagrams into the correct order for processing in upper layer protocols. The re-assembly of these data units into the correct order at the end system is one of the most severe bottlenecks to high throughput of single session communications. The effect of an increased "load" on this process that may be incurred by such an "inverse multiplexing over IP" has not been studied.

2.7.2 DES ASIC

2.7.2.1 Overview

DES is a well-studied, heavyweight algorithm for protecting communications and computer data. The Sandia National Laboratories (SNL) DES Application Specific Integrated Circuit (ASIC) [87], an implementation of the Data Encryption Standard algorithm as defined in FIPS Pub 46-2 [20], is a high-speed, fully pipelined implementation providing encryption, decryption, unique key input, or algorithm bypassing on each clock cycle. In other words, for each clock cycle, data presented to the

ASIC may be encrypted or decrypted using the key data presented to the ASIC at that cycle or the data may pass through the ASIC with no modification. Operating beyond 105 MHz on 64 bit words, this device is capable of data throughputs greater than 6.7 Gbps, while simulations show the chip capable of operating at up to 9.28 Gbps. In low frequency applications the device consumes less that one milliwatt of power. The device also has features for passing control signals synchronized to the data.

The SNL DES ASIC was fabricated with static 0.6 micron CMOS technology. Its die size is 11.1 millimeters square, and contains 319 total pins (251 signals and 68 power/ground pins). All outputs are tri-state CMOS drivers to facilitate common busses driven by several devices. This device accommodates full input of plain text in 64 bit blocks, and a complete DES key of 56 bits. Additionally, 120 synchronous output signals provide 64 bits of cipher text and the 56 bit key.

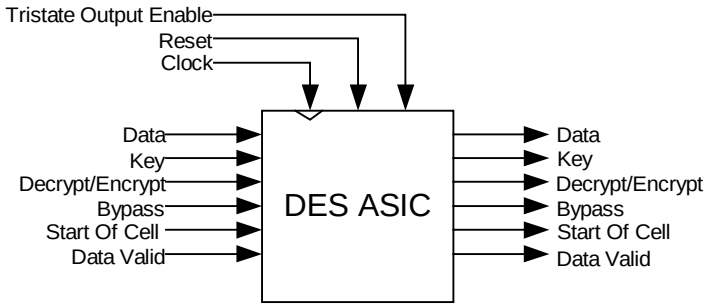


Figure 10. DES ASIC Block Diagram

Three input only signals (Figure 10) control electrical functions for logic clocking (CLK), logic reset (RST), and the tri-state output enables (OE). The CLK signal provides synchronous operation and pipeline latching on the rising edge. Both RST and OE are asynchronous, active high signals.

Two synchronous signals, decrypt/encrypt (DEN) and bypass (BYP), determine the DES cryptographic functionality. On the rising edge of each CLK, the logic value presented to the DEN input selects whether input data will be decrypted (logic 1) or encrypted (logic 0). In a similar manner, BYP selects algorithm bypassing (logic 1) or not (logic 0) for each clock cycle. Both of these signals pipeline through the ASIC and exit the device synchronous with the key and data.

Two more signals, start-of-cell (SOC) and data valid (VAL) enter and exit the device synchronous with data and key information. These are merely data bits that may provide any user-defined information to travel with input text and key. These signals are typically used to indicate the start of an ATM cell and which words in the pipeline contain valid data.

2.7.2.2 Design

The DES algorithm was implemented using VHDL and synthesized into the Compass library of standard cells. The device (Figure 11) was fabricated in Sandia's MDL (Microelectronics Development Laboratory). Two wafer lots were successfully fabricated. (Although the ASIC design and fabrication were DOE/ASCI funded efforts, subsequent efforts to package and integrate the ASIC into high speed communication systems were funded under this LDRD.)

ASICs from both wafer lots were shown to operate beyond the maximum frequency (105 MHz) of Sandia's IC Test systems. For 64-bit words, this equates to 6.7 Gb/s. This operational frequency was tested over a voltage range of 4.5 to 5.5 Volts and a temperature range of –55 to 125 degrees C.

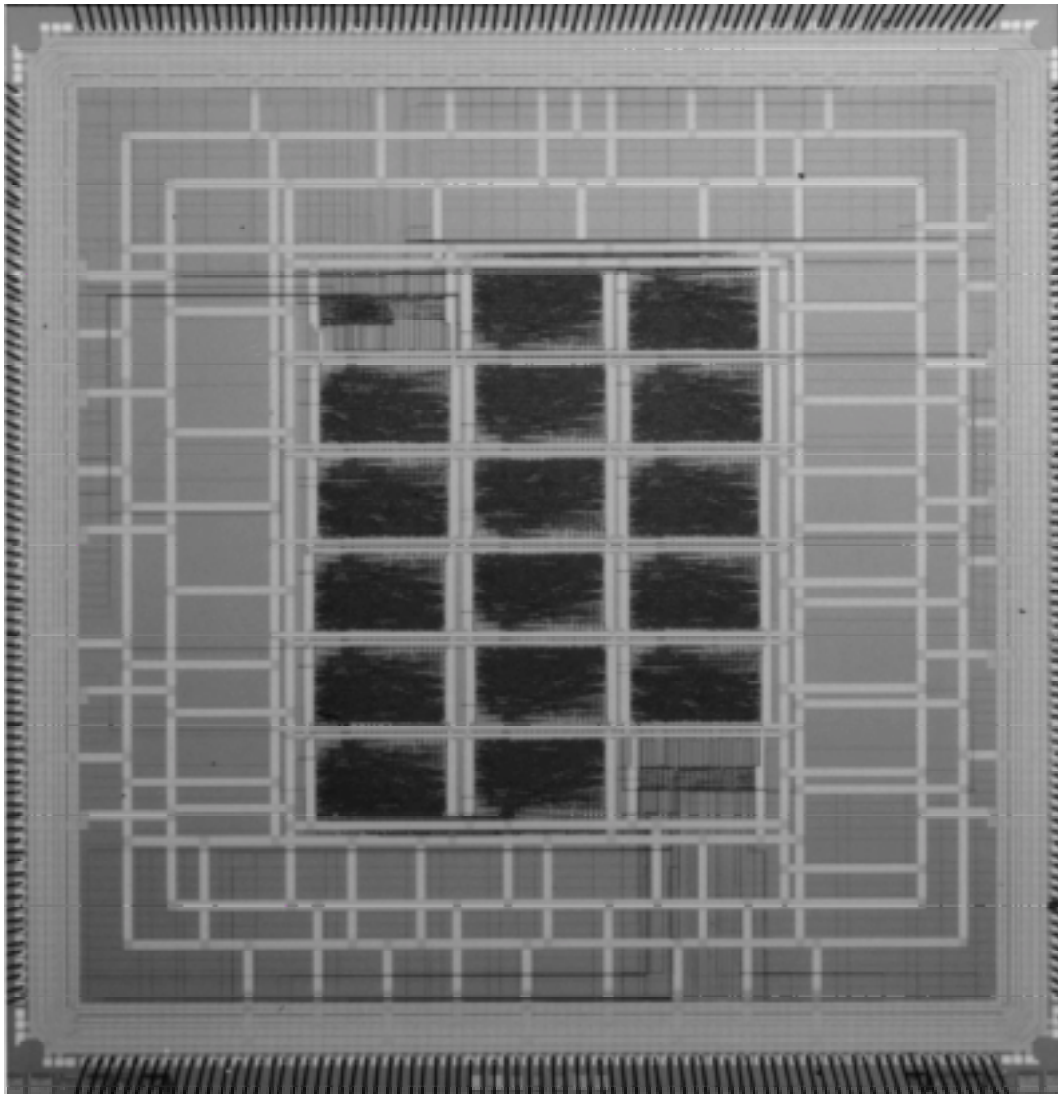


Figure 11. DES ASIC Die (11.1 x 11.1 mm).

This implementation is a fully pipelined design. It takes eighteen clock cycles to completely process data through the pipeline causing the appropriately decrypted, encrypted, or bypassed data to appear on the ASIC outputs. Additionally, all key and control input signals pass through the pipeline and exit the ASIC synchronized to the ciphertext outputs.

Pipelining the DES algorithm increased the device throughput by dividing the algorithm into equally sized blocks and latching information at the block boundaries. This gives signals just enough time to process through each block between clock cycles, thereby maximizing the operational frequency.

Pipelining the algorithm also allows a high degree of key and function agility for this device. Here, agility means that the SNL DES ASIC processes inputs differently on each clock cycle. As an example, the device may encrypt data with one key on one clock cycle, decrypt new input data with a different key on the very next clock cycle, bypass the algorithm (pass the data unencrypted) on the following clock, then encrypt data with yet another independent key on the fourth clock cycle. The control signals used to select these various modes of operation are presented at the output, passing through the device synchronized to the input data and the input key information. All inputs and outputs (control, key, and data) enter and exit the part synchronously. This per-cycle input and output of all variables facilitates cascading the devices (to implement Triple-DES) for increased encryption strength, and paralleling the devices for even higher throughput.

2.7.2.3 Packaging Issues

The SNL DES ASIC has been packaged into three different packages including a 360 pin PGA, a 503-pin PGA and a 352 pin BGA. The original 360-pin package was used in initial testing of the DES ASIC performance. It was in this package that the DES chip was shown to operate at over 105 MHz. Sandia had earlier developed a 1.1 million gate PLD board [64] that used 11 Altera 10K100 devices. This board was used in the development of the DES ASIC pipeline design, housing the four 10K100 devices. It was determined that the SNL DES ASIC could be used with the original PLD11 board, being

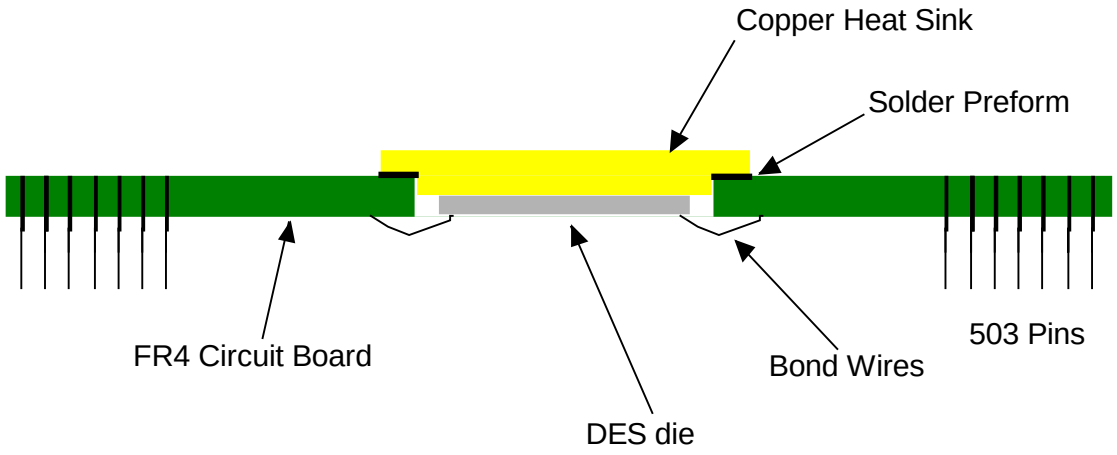


Figure 12. Cross Section of the 503-Pin Package.

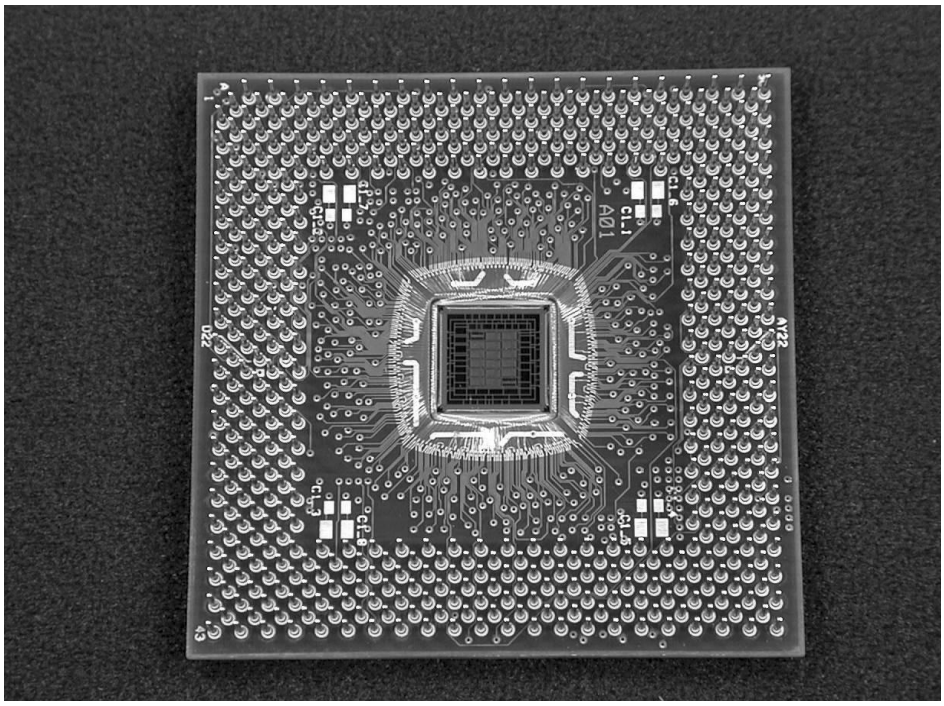


Figure 13. Picture of the 503-Pin FR4 Board Package.

substituted for a single 10K100 device, if a 503-pin equivalent package were available. Sandia designed an FR4 board onto which the DES ASIC was wire bonded and 503 pins could be inserted. The chip-on-board package had to be designed to dissipate up to 5 watts produced by the DES ASIC. This is accomplished by attaching the DES die directly onto a gold plated copper insert that is attached to the FR4 board using a tin-lead solder preform. Pictures of a representative cross section and this package are shown in Figures 12 and 13.

The design of this package enables a heat sink and integrated fan to be attached to the back of the copper insert to enable the package to dissipate over 6 watts. The FR4 printed wiring board uses 3 mil copper traces and spaces with 5 mil vias. This design also allowed the board to be used to connect the existing bus signal assignments from the PLD11 board to the appropriate key and text signals on the SNL DES ASIC. Two versions of the package were designed and fabricated. Each has a different wiring schematic designed to fit into a different socket on the PLD11 board. SNL DES ASICs in the 503 pin package were demonstrated in November 1998 at the Supercomputing '98 Conference in Orlando, Florida.

The SNL DES ASIC has also been packaged in a 35 x 35 mm, 352-pin ball grid array (BGA) package. This is an open tool commercial package available from Abpac Inc. (of Phoenix , Arizona, U.S.A.). The package was chosen not only for its capability to dissipate over 5 watts and smaller size, but also its low cost. Abpac's automated

manufacturing capability enabled a reduction of over 20 times in packaging costs. This package is being used in the design of a triple key, triple DES encryption module.

2.7.2.4 Redesign Considerations

As built, the SNL DES ASIC has successfully demonstrated how a "heavyweight" encryption algorithm can be accelerated into the 10 Gbps arena. However, there are several enhancements that could have improved performance, device yield, and upper level design and integration. These enhancements should be considered for any future encryption ASIC, either a follow-on DES or AES. Performance modifications include the chosen ASIC fabrication technology, length of the pipeline, the number of residing encryption engines, and I/O (input/output) buffers. Device yield could have been improved by shrinking the silicon area. Finally, the next-level integration would have benefited from modifications to the input and output signals and inclusion of certain key management operations into the ASIC.

Sandia National Laboratories MDL CMOS VI (0.6 um CMOS) was used to fabricate the original SNL DES ASIC. This process was chosen for economy, locality, familiarity, and ease of use. Fabricating with this process imposed an upper bound on the ASIC throughput at 9.7 Gbps. For maximum throughput a newer, faster process would have reduced the pipeline delays. Any future design effort should compare the costs and benefits associated with Sandia's CMOS VII (0.35 um CMOS) against foundries processing smaller, faster features. Smaller feature sizes would offer additional opportunities to improve the design architecture.

Architectural modifications would improve the performance with shorter pipeline delays, parallel encryption engines, and faster I/O buffers. The original SNL DES ASIC was built with sixteen pipeline stages for the algorithm. Increasing the number of pipeline stages, or increasing the pipeline length, would distribute the combinatorial logic gates and reduce the minimum delay between registers. This would result in a greater data throughput at the cost of increasing the clock latency for any given datum. Fabrication with smaller features would also allow greater gate density and the possibility of parallel engines simultaneously processing distinct data. Further throughput enhancements would be realized by replacing the CMOS-level (0 to 5 Volts) I/O buffers with faster buffers such as LVDS (low voltage differential signal) interfaces. The CMOS buffers exhibit an upper frequency limit of 200 MHz, while LVDS work beyond 600 MHz.

Device yields on the SNL DES ASIC were reduced because of the large die area in silicon. This area could be reduced, by running two rings of I/O bondpads, rather than a single I/O ring. Reducing this area would produce more die per wafer and lower the losses-to-silicon-defect density. The alternative would be to fill the unused area in silicon with more functionality, so the cost of low yield would be offset with the increased functionality.

At the next-level assembly, the SNL DES ASIC suffers from having the wrong flavor on some output signals. This is apparent in Triple-DES configurations of three cascaded

devices. In any similar, future design it would be wise to invert the output of the encrypt/decrypt bit for cascading. Ideally, it would be nice to make signals of this nature programmable. Also, the outputs of the SNL DES ASIC were tri-state drivers to allow the possibility of ping-ponging two devices for enhanced throughput. In retrospect it would be better to use simple output drivers that are faster than the tri-state outputs.

2.7.3 Data Compression

Data compression algorithms can be categorized into two groups: lossless and lossy. Lossless compression techniques are those guaranteed to generate an exact duplicate of the input data stream after a set of compress and decompress/expand operations [47]. This is the type of compression used in storing or transmitting records and files where complete accuracy is required, and losing any bits would be an error and may possibly have serious consequences. Lossy compression techniques give up a certain amount of accuracy in trade for increased compression ratios. These are suitable for digitized voice, and usually, graphic images. If the amount of loss is too great, artifacts are introduced, which reduce quality and render the data less usable.

2.7.3.1 Lossless Data Compression

Linear recurring sequence generators expand a small seed of information into a very large pattern (similar to a decompression operation). The efficiency with which this can be done led to consideration of using linear sequences for high speed, high ratio compression.

Prior work has shown that the operation of linear recurring sequence generators in the form of linear feedback shift registers can be readily accelerated by generation of these sequences in parallel [59].

Basically, a linear recurring sequence is a sequence of bits, 2^n-1 in length, that can be represented and generated by an n bit polynomial starting from an n bit state. For these particular sequences, an extremely large compression ratio of (2^n-1):2n can theoretically be achieved.

An obvious question here is why we would think that LFSRs would provide compression in the first place. The answer is that we were following up on a suggestion made by Massey himself, one of the developers of the Berlekamp-Massey LFSR synthesis algorithm [42]. We could find no mention in the literature of Massey's suggestion having been put to the test. So we developed an implementation of the algorithm in the C programming language [14], then proceeded to test it.

Reduction of this method to practice involved methods of describing and communicating arbitrary sequences as "piecewise linear" and/or the differences between an arbitrary sequence and a "piecewise linear" description.

We showed that for general input sets, linear feedback shift registers (LFSRs) do not provide compression comparable to current, standard algorithms, at least not on the current, standard input files. Rather, LFSRs provide performance on a par with simple, run-length encoding schemes. We exercised three different ways of using LFSRs on the Canterbury, Canterbury (large set), and Calgory Corpora, and on three, large graphics files of our own. Details of the research were reported elsewhere. [15]

2.7.3.2 Lossy Data Compression

There are typically three tradeoffs that must be considered in selecting a mechanism to support remote display of high-resolution visual output from a high performance computation system. These are: image quality, interactivity, and network bandwidth required. Most solutions select high image quality at the expense of interactivity and/or network bandwidth. To deliver a 1280x1024 pixel, 30 frame per second, 24 bit color display (without compression) would require approximately 1 Gigabit per second! A remote visualization system, made of off-the-shelf components, was prototyped at Sandia using Quadrant Processing with data compression.

In this prototype, a visualization server splits the 1280x1024 framebuffer into four 640x512 quadrants or "display heads", which are individually converted to NTSC video and encoded in parallel using scan converters. The output from each scan converter is fed through a Fore Systems video compression device, where it is operated upon by a JPEG compression algorithm. The compressed data is then sent over an ATM network to its destination.

At the destination, the video signals are, in parallel, decompressed, and converted back to NTSC video. The four signals are then recombined into one high-resolution RGB signal using an RGB Spectrum multiple video window display system (such as used for video surveillance systems). At this point the RGB signal can be displayed on a monitor or projected on to a "powerwall" or other large format display.

This technique of splitting the display into four quadrants and re-combining the quadrants at a destination client, was generally successful. Some calibration effort was required to configure the video recombiner to properly align each of the quadrants, and to properly match colors across all quadrants. Using a subjective assessment of image quality from preliminary users, several aspects of the system were investigated, including image quality vs. bandwidth requirements. Results indicated that with the least amount of compression, good image quality was still achieved while only consuming 6 Mb/s of bandwidth per video signal (24 Mb/s overall). Further details and results regarding this remote visualization method were published in a paper by Friesen and Tarman [26].

Ongoing tests include examination of other compression algorithms (Motion JPEG, MPEG-2, wavelet compression, and MPEG-4) to determine which one provides the best bandwidth vs. video quality tradeoff.

2.7.4 Standards Bodies Activities

Sandia National Labs has been very active in the standards groups as part of the 10-100 LDRD. Sandia promotes rapid maturation of high speed network and chip-to-chip standards in order to see the early introduction of commercial equipment that will meet the needs of the DOE/DoD WAN community. As little as three years ago (circa 1997) there was little work going on in the standards arena for standards at 2.5 Gbps and above. The ATM Forum (ATMF) was struggling with a newly introduced 2.5 Gbps UTOPIA 3 standard. The OIF (Optical Internetworking Forum) was still newly formed and was yet to tackle the high speed standards. Sandia personnel worked to get the UTOPIA 3 standard finalized and approved. Sandia, in conjunction with General Dynamics, AMCC and other DoD interests introduced a working document in the ATM Forum for the yet to be defined UTOPIA 4, 10 Gbps specification. Over the next year and a half, Sandia worked to make this standard a reality. This specification was unique at the time in that it allowed for the passing of both ATM Cells (specified by 53 byte blocks of data) and variable length data packets (more like TCP/IP formats). The result was the UTOPIA Level 4 specification, approved in the Spring, 2000.

The Utopia 3 Physical Layer Interface specification is af-phy-0136.000 and can be found on the ATM Forum web site at ftp://ftp.atmforum.com/pub/approved-specs/af-phy-0136.000.pdf. The UTOPIA Level 4 specification is af-phy-0144.001 and can be found at ftp://ftp.atmforum.com/pub/approved-specs/af-phy-0144.001.pdf.

2.7.4.1 UTOPIA 3

The ATM Forum's Level 3 interfaces are the result of considerable debate between two competing designs for operation of an interface at 2.5 Gbps. Work in the physical working group on the standard continued for over a year. With the details of the primary standard completed (called UL3), alternative standards for frame-based (packet) extensions were proposed, two originating within the ATM Forum, and one from an industry group of chip manufacturers. Seeing that the vote for an ATM Forum version of the standard could not be pushed forward to a final ballot, the other standard was proposed and accepted. This standard became Frame Based ATM Interface (Level 3) (FBATML3) and can be found at ftp://ftp.atmforum.com/pub/approved-specs/af-phy-0143.000.pdf.

UL3 can have either an 8, 16, or 32 bit interface operating at a maximum speed of 104 MHz It specifies the transfer of fixed length cells between single or multiple PHY ports. The FBATML3 interface is not directly compatible with the UL3 interface. FBATML3 permits either 8 or 32 bit widths, also operating up to 104 MHz. This bus interface is point-to-point (not supporting multiple PHY ports). The logical data words are bytes (or double word, 4 bytes) that accommodate variable packet sizes as well as fixed length cells. The FBATML3 interface uses an external clock that is presented to both the receiver and the transmitter interfaces. Seven interface control signals are used to identify packet beginning and end. These signals represent out of band signaling. Eight other

signals are used to send FIFO status information. This interface is referenced other places as POS-PHY (level 3).

2.7.4.2 UTOPIA 4

The UL4 interface was developed around the idea of using in band FIFO control. Since almost all physical layer interfaces are bi-directional, it was reasoned that FIFO status control could be performed by sending this information in the return data path, between data packets. This functionally makes the interface quite different than others previously proposed. The interface has a one bit mode that can be used for FIFO status if only a unidirectional interface is required. In this mode, this interface uses less data lines than some other standards. The basic UL4 interface operation was first drafted by Perry Robertson (SNL) and Dave King (General Dynamics).

The UL4 interface consists of 32/16/8 Low Voltage Differential Signaling (LVDS) data bit pairs, a control bit pair and a clock bit pair. The total pins required to operate a 32 bit UL4 interface is 68 in any one direction. There are three options for the data path width, 32 bits, 16 bits or 8 bits. The interface operates at a nominal rate of 415 MHz for each, therefore, the 8 bit version is suitable for 2.5 Gbps operation. The interface is nominally symmetric having the same interface specification on the transmit and receive side of the interface. However, the interface can be implemented as little as a one bit interface. In this case, the single bit return path acts like a FIFO status control signal allowing the interface to stop the flow of data to the receiver. The data is clocked on the rising edge of the clock.

2.7.4.2.1 In-Band Control

The UL4 specification implements flow control as in-band control words. In other words, addressing, and start and stop commands are inserted into the data stream. If flow control is to be implemented, then a return data path must be implemented. The control bit is used to differentiate a data word and a control word. Each word is a logical 32 bits no matter what physical data path width is used in the implementation.

There is still considerable disagreement as to the benefits of having flow control on the physical interface. It is not clear to all camps that flow control serves any tangible purpose. In their system designs, if the receiver cannot handle the incoming data (due to full buffers) the data must be discarded (tossed onto the floor as it were). The reality, they would say, is that the only purpose of this interface is to pass data and that the only way the flow of data could be stopped up (at the exit) was for some other problem in the system to have occurred, and to have the physical device drop data packets or cells would be acceptable behavior. There is not any good way for a device to stop data coming in from a long haul fiber, for instance. It is sure that this issue will reappear in discussions that are taking place for the development of the UTOPIA 5 standard for OC768 (40 Gbps) communications.

2.7.4.2.2 Frame Based ATM Level 4

As part of the compromise to get UL4 passed, an additional work item was begun in the ATM Forum called Frame Based ATM Level 4 (FB4). As with the Level 3 specification, there were two competing designs for this 10 Gbps interface. An alternative interface, largely developed by a group of chip vendors, was being proposed in the OIF as the OC-192 standard interface (called SPI-4, for System Packet Interface, Level 4). The ATM Forum has worked to insure that this standard will interoperate correctly and is near approval of this standard. This interface will become the first standard that is essentially the same in both the OIF and the ATM Forum.

2.7.5 Prototype Development

2.7.5.1 DES ASIC Encryption Demonstration

An encryption demonstration was assembled using the SNL DES ASIC [87] and the Sandia-developed PLD-11 board [64]. It was used at Supercomputing '98 in Orlando, Florida, and at Sandia's Family Day in 1999 to showcase this technology.

The demonstration used an Altera 10K100 programmable logic device (PLD) to generate a bit pattern. This bit pattern was displayed by a panel of light-emitting diodes (LEDs), which plugged into the PLD-11 board. The block of bits was then routed to a DES ASIC where it was encrypted and displayed by another LED panel, thus showing the pseudo-random pattern expected by encrypting the data. From there the data was routed through another PLD to perform some re-ordering of the data lines in preparation for decryption. The data blocks were then decrypted with another DES ASIC and displayed with another LED panel, showing the original bit pattern, delayed by 36 clock cycles.

This demonstration was operated at clock speeds ranging from 0.42 Hz (about 27 bps) to 5 MHz (about 320 Mbps). In the lab, this demonstration operated at clock rates up to 80 MHz, corresponding to a data encryption/decryption rate of 5.12 Gbps.

2.7.5.2 OC-48 Bit Error-Rate Tester

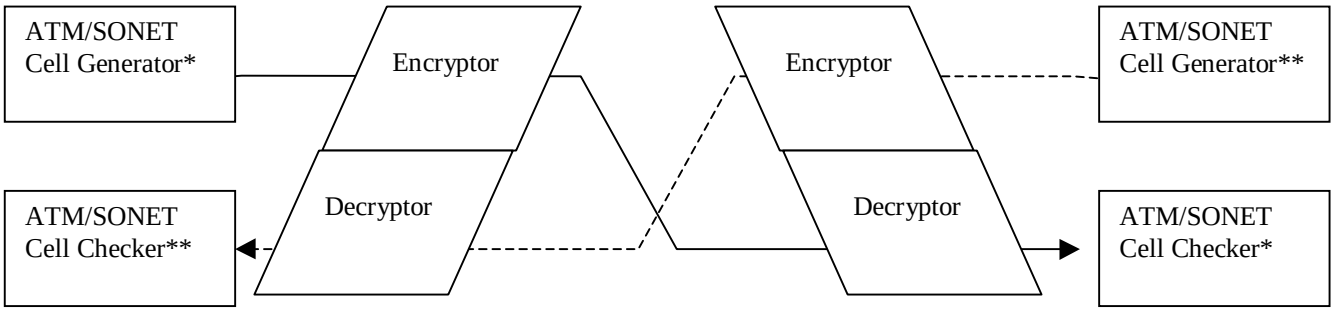
Sandia National Laboratories has developed technology with which to scale the generation and checking of patterns useful for testing extremely high speed communication links. This technology has been disclosed in two Sandia Technical Advances entitled "Efficient Synchronization of Bit-Error Communication Test Equipment" [58] and "Parallel Generation of Linear Recurring Sequences" [60].

The improved synchronization method simplifies the circuitry required to achieve synchronization, scales for ultra-high speed operation, and is easily applied to arbitrary width sequence generators and to parallel word framing of arbitrary widths. The method of Parallel Generation of Linear Recurring Sequences not only scales for high throughput, but also identifies a method for choosing the appropriate polynomial to

minimize the hardware required to compute the next set of parallel bits, optimizing the maximum frequency of operation (f_{max}).

These advances have been embodied in several (SNL) laboratory test devices, most notably an 155 Mb/s OC-3 ATM/SONET Cell Generator/Checker (circa 1994), and a 2.5 Gb/s OC-48c ATM/SONET Cell Generator/Checker (developed under this LDRD).

In order to perform rigorous testing of communication channels, pseudorandom bit patterns are generated at the transmitter. The received bit pattern is compared with the same pseudorandom bit sequence (generated independently at the receiver) for correctness. Because there is some communication delay between the transmitter and the receiver, some method of synchronizing the receiver pattern generator with the delayed transmitter generated pattern is required. This Bit Error-Rate Tester (BERT) could be employed as shown in Figure 14.



The generator/checker pair marked with the (*) will be interconnected to the generator/checker pair marked with the (**) for round-trip cell latency measurements.

Figure 14. In-circuit Configuration of Bit Error-Rate Test Equipment.

The OC-48 BERT uses the SNL developed PLD-11 board [64] as a host, supplying physical structure and power. A mezzanine card (developed by Sandia) contains several Altera 10K50 PLDs and a Tektronix OC-48 interface, operating at 2.5 Gbps. One PLD contains the linear recurring sequence (LRS) generator. The other PLD contains a copy of the same linear recurring sequence generator, the checking logic to compare the incoming sequence with the generated sequence, and the synchronization logic. In general, a bit pattern is generated, built into an ATM cell, and sent to the OC-48 interface where it is framed into a SONET payload and transmitted. When a SONET frame is received, it is operated upon and resulting ATM cells passed out of the Tektronix board to one of the PLDs on the mezzanine card. The payload is extracted from the ATM cell and compared against the bit pattern generated by the second LRS generator.

The OC-48 BERT operates with a pseudorandom pattern of period 2^{127}-1 bits (approximately 1.7×10^{38} bits). This pattern is inserted into 384 bit cell payloads, resulting

in $(2^{127}-1)*384$ cells before the pattern repeats in the cell payload window. At OC-48c, this period is approximately $6.5x10^{40}/2.5x10^{9} = 2.6x10^{31}$ seconds or $8.3x10^{23}$ years.

2.7.5.3 OC-48 Encryptor

The PLD11 reconfigurable logic platform was used to demonstrate the operation of the DES ASIC on real communication data streams. Though the PLD11 board [64] was developed to prototype and investigate the processing of communication functions at 10 Gb/s and beyond, the fastest I/O modules available at the time of this work operated at 2.5 Gb/s. Therefore, a Tektronix OC-48c ATM/SONET framing circuit module was adapted to pass 2.5 Gb/s data streams into and out of the PLD11. A circuit board developed to perform this adaptation of signals between the 2.5 Gb/s ATM/SONET framer and the PLD11 is called simply the "mezzanine" board.

2.7.5.3.1 Mezzanine Board

The Mezzanine board was connected to the Tektronix ATM/SONET framer with a UTOPIA 3 interface [25]. (The Tektronix interface was implemented slightly before the UL3 standard was finalized and is technically a "UL3-like interface", but the differences were inconsequential to this work.) The interface between the Mezzanine board and the PLD11 was a 128 bit wide interface developed and proposed for standardization as the UTOPIA 4 (10 Gb/s) interface. Subsequently, other interface widths were considered and the UL4 was standardized on a 32 bit interface, but with the incorporation of several "lessons learned" from this 128 bit wide prototype.

2.7.5.3.2 Data Path

The data path between the 32 bit wide UL3 interface and the 128 bit wide interface to the PLD11 incorporated a 32 bit to 128 bit de-multiplexer and a state machine. The state machine controlled the packing of 32 bit words into two 64-bit parallel encryption paths, and also replicated the header information into both paths. The replication of header information simplified the cryptographic context lookup by allowing each cryptographic path to do an independent cryptovariable context lookup, retrieving the cryptographic key on the basis of the virtual circuit identified in the header.

2.7.5.3.3 Clock Distribution – Source Synchronous Clocking

Printed wiring boards have traditionally been designed with a central clock distribution system that delivers the same clock reference signal to each integrated circuit component. Due to the difference between the distance traversed by the data path and the clock path , the clock/data phase relationship must be compensated at each integrated circuit. This technique requires the clock paths to be re-phased whenever the operating frequency is changed significantly, since the phase relationships over different propagation distances change with frequency. While the PLD11 was designed with specialized "central" clock distribution and the ability to compensate the phase of each clock signal delivered to each

major integrated circuit component, another method called “source synchronous clocking” was used.

Source synchronous clocking simply involves the clock, data, and associated control signals traveling the same physical path (subject to the same propagation delay) as data is processed by successive components. This minimizes clock and data skew, and ensures that the phase relationship between clock and data remains unchanged at the signal destination. If the clock and data do not travel over the same delay path, a change in frequency will result in a change in the phase of data with respect to clock at the signal destination. Non-source synchronous designs must be compensated for operation at a single frequency. Source synchronous designs need not be compensated for operation over a wide range of frequencies. This technique works for point-to-point and even for point-to-multipoint circuits. Since bi-directional signal busses have more than one “data origin”, source synchronous clocking does not apply to shared buss structures.

This method was used successfully to operate the encryption data path over a wide range of frequencies. One particular advantage of this method is the ability to slow the clock rate down to the range of “seconds” so that switching phenomena can be observed on LEDs with the human eye (for debugging purposes).

2.7.5.3.4 Data Width

There are several design considerations in support of the extremely wide 128 bit data path. These considerations include 1) clock rate for data transfer; 2) pin cost and reliability; 3) data skew; and 4) flow control between circuit segments using different clock signals.

The clock distribution circuitry on the PLD11 was optimized for operation at about 20 MHz. In order to transfer 2.5 Gb/s with a word transfer clock rate of 20 MHz, a transfer width of $2.5 \times 10^9 / 20 \times 10^6 = 125$ bits is required. This bit width was rounded up to 128 bits (a multiple of 32 bits, in order to simplify the demultiplexing/multiplexing circuitry). The encryption circuitry was then made to clock at 25 MHz so that the encryption/decryption process would always “outrun” the data input channel, thereby simplifying the flow control at the boundary.

Data flows in parallel into and out of multiple integrated circuits. The more pins in each path, the more likely that the data bit path will be interrupted by circuit integrity problems such as pin-to-pin mating difficulties in a connector, or poor solder joints between package pins and circuit board traces, or even solder bridge shorts between adjacent pins. In addition, the more data pins in the path width, the more likely that the path integrity of one or more data bits will be interrupted at some point in the data path.

The “per pin” cost of each additional pin in an integrated circuit package and connector pins and data bit path traces on circuit boards is not large, but do add up quickly to larger costs. This is less of a consideration for “research prototypes” built in small quantities,

but is a large consideration for commercial devices, competitively marketed in large quantities.

Subsequently to this research design, IC designers have discovered that it is more cost effective to increase the clock rate (internal to the integrated circuit package) and therefore reduce the package pin count for high throughput designs. Designs proposed to the ATMF for 10 Gb/s chip-to-chip data transfer ranged from 128 bit width at 80 MHz down to 16 bit width at 800 MHz, and over the course of the UL4 effort, the 32 and 16 bit widths were chosen for standardization.

Currently, integrated circuit packages for First In First Out buffers are commonly available in 18 bit widths (again, to optimize pin cost and reliability). To provide smooth and error free flow control on the extremely wide (128 bit) data path, multiple (8) FIFO integrated circuits were employed.

Within an single Integrated Circuit FIFO package, the circuit dimensions and signal integrity over multiple "on-chip" paths allow the multiple input bits clocked on a single clock edge to remain associated with each other and to be reliably output together on a single clock edge.

However, poor signal integrity on the clock and enable lines shared between the parallel FIFO packages can cause loss of synchronization between the multiple FIFOs. That is, the eight 16 bit words latched into the FIFO inputs on the same clock cycle may not all become output on a single clock cycle. To prevent this, the clock and enable signals for both input and output of the FIFO must be carefully generated, distributed, and terminated, so that these signals appear the same to each of the multiple FIFOs. In addition, a spare FIFO signal must be inserted into each FIFO periodically, and monitored for synchronization across the multiple FIFOs, in order to detect synchronization loss. When mis-alignment of these synchronization signals is detected at the output, the entire set of FIFOs must be reset to accomplish re-synchronization. This additional circuitry requires an additional feedback signal from the output process (frequently on a separate circuit board) to the reset input on the FIFO circuitry.

2.7.5.3.5 Monitor/Debug/Lessons Learned.

In order to diagnose signal integrity problems in the data path, some means of attaching a logic analyzer to each pin in the path must be provided, and data patterns designed to easily spot discrepancies in the processing of data path bits must be inserted on the data paths. These patterns must be designed to detect common difficulties such as open bits and shorted bits (such as a circulating "1" in a field of zeroes, and a circulating "0" in a field of ones). The reconfigurable logic devices facilitated the hardware compilation of different pattern generators in the data path. In the Tektronix/Mezzanine/PLD11 design, the 128 bit path between the Mezzanine and PLD11 boards was well instrumented with controlled impedance connectors for mass termination of logic analyzer probes, but the data path input to the FIFOs on the Mezzanine board was not as well instrumented. This required the attachment of individual logic analyzer signal lines to package pins with

micro-clips in some places. In future designs, greater attention should be paid to methods of "mass termination" of logic analyzer probes using miniature controlled impedance connectors (such as MICTOR connectors).

3 Roadmap to 40 Gbps

The path to 40 Gbps network communications will encompass the following research areas:

- Streamlining the communication protocols, including the use of end-to-end theoretical modeling, inverse multiplexing, increased parallelism, and OS Bypass architectures (e.g. VIA, ST, AGP – Advanced Graphic Port);
- Utopia interfaces (how to communicate 40 Gbps from one chip to another on a circuit board or through a daughtercard connector.);
- Accelerating supporting functions, such as encryption, authentication, and data compression.

3.1 Streamlining the Protocols

Ancillary to this development has been work to streamline the communication protocol processing required to "access" the tremendous bandwidths represented by multiple wavelengths. Streamlining WDM access has involved various combinations of "bypassing" the processing and functionality provided by various combinations of the IP, ATM, and SONET protocol layers. This has led to speculation that ATM Services and even IP Services may cease to be available from long haul service providers in the later part of this decade, even though the ATM switching market is healthy and growing [3], and the IP routing market is even larger and growing even faster.

Two of the general trends in this area are worth examining further. Advances in the communication performance of processes implemented in silicon follow Moore's law, doubling every 18 months. Optical performance improvement doubles every 12 months [84]. The widening gap between optical performance and silicon performance implies that ever higher layers of protocol processing should be done in the optical domain to achieve high speed, especially if the underlying transport technology is optical.

The very factors that have allowed this phenomenal growth have also changed the basic economics of the industry. With each passing year, voice service growth trails significantly behind bandwidth-hungry IP data growth. Carriers, naturally interested in capturing their share of the Internet market while converging legacy services in the process, are now deploying switches and routers developed by Internet data-centric companies (e.g., Juniper, Cisco) and established telecommunication equipment providers (e.g., Lucent, Nortel). These companies are responding to the needs of their customers. As is evident by reviewing their product offerings for data services, Asynchronous Transfer Mode (ATM) is not being offered as a high-speed option (indeed, there's even a lack of OC-48 ATM support in these new products). Rather, they are developing high-performance switch/routers that use frame-based data link technologies in an attempt to exploit better DWDM's "big pipes" by reducing protocol overhead, frame conversion complexity, and equipment cost.

3.2 ATMF UTOPIA 5 Work Item

On the heels of the UL4 specification, the ATM Forum began work on an OC-768 physical interface to be called UTOPIA Level 5. Work has also begun in the OIF on a similar standard called SPI-5 (System Packet Interface, level 5). Currently, there are leanings in the industry for a standard that looks somewhat like the FB4 interface with appropriate increases in clock rate. The UL5/SPI-5 40 Gbps interface specification will be subject to the ongoing "tug of war" between desires for smaller pin counts and the desire to keep clock frequencies at manageable rates.

3.3 Encryption

3.3.1 Accelerating Counter Mode Encryption

Counter Mode encryption can be accelerated to 40 Gbps by using a variety of techniques. They include pipelined encryption engine architecture, smaller and faster feature geometries in the ASIC production process, high speed ASICs from a SiGe or GaAs production process, and multiple encryption engines operating in parallel on pieces of the data (cell or packet payload),

3.3.2 Accelerating Cipher Block Chaining Mode Encryption

Because of the feedback loop in the cipher block chaining encryption mode, CBC encryption is difficult to accelerate. When a pipelined encryption engine is being used to perform CBC mode encryption, the pipeline must be "flushed" or "run dry" before a ciphertext value is obtained and fed back to Exclusively-OR with the next block of plaintext. There are methods described elsewhere [65][66] of gathering multiple sessions together to keep a pipelined encryption engine running at full rate.

3.3.3 Accelerating Authentication

The question that we have pursued here is the following: is it possible to provide robust authentication without a feedback path? CBC-MAC, for example, uses a feedback path: authentication of block i cannot begin until authentication of block i-1 is complete. This feedback path precludes pipelining and complicates parallelization of authentication. Ideally we would like to be able to use the same acceleration approach for authentication that is typically used for encryption, namely pipelining. However, if robust authentication is not possible without a feedback path, then pipelining is not possible for authentication.

An alternative is to use what Schneier [71] calls "interleaving:" input blocks 1, 2, ..., n are fed to authentication units 1, 2, ..., n, respectively, and when they are done, input blocks 1+n, 2+n, ..., 2n are fed to authentication units 1, 2, ..., n, respectively, and so on. The message authentication code increases by a factor of n, but this approach does provide for some parallelization. However, it was our intent to find an algorithmic solution.

Of the four possible DES modes of operation—ECB, CBC, CFB, OFB—only CBC is suitable for authentication. ECB authenticates only the last block, unless each processed block is part of the authentication code, in which case we would be authenticating by sending the cleartext, followed by the encrypted text. CFB, like CBC, requires feedback. OFB is promising, since the cipher stream can be prepared in advance, enabling parallel processing, but, like ECB, OFB authenticates only the last block. [23]

We searched for evidence that feedback is not required for authentication. We found few algorithms that support this approach. One approach, published recently at the NIST AES "Modes of Operation" conference by Charanjit S. Jutla (IBM) introduces an independent variable in lieu of feedback [33]. Another approach is called the XOR MAC algorithm, developed by Bellare et al. [5], and is of interest "particularly for high-speed networks," as the developers note. This algorithm provides authentication without requiring feedback, thereby enabling pipelining. The algorithm is based on finite pseudorandom functions (PRF). A PRF can be defined from a block cipher (such as DES), a hash function, or a compression function. Without feedback, blocks can be processed independently of each other. The algorithm works as follows.

A (possibly padded) message is broken into fixed size blocks. Each block is concatenated with a 1 bit and the number of the block. (The block number is represented by the same number of bits as the block itself.) That concatenated set of bits for each block is then run through the PRF. An additional block is also run through the PRF. This additional block has the same number of bits as the other blocks: it contains a 0 bit and a random number, represented by twice the bits as in the fixed size blocks. All of the blocks are then XORed together. The resulting MAC is a pair where one element is the result of the previous step and the other element is the random number used.

Again, note the absence of feedback in the algorithm. For an n-block message and a machine with n functional units each capable of running a chosen PRF, one could run PRF on all of the blocks in the first step, followed by running all of the XORs in log(n) additional steps.

The above scheme is referred to as "randomized" XOR, indicating the use of a random number. An alternative scheme, referred to as "counter-based" XOR, calls for the sender to maintain a counter that is incremented for each message authenticated. The counter in the counter-based scheme replaces the random number in the randomized scheme. The counter-based scheme is more secure than the randomized, and both are "more secure" than CBC MAC, according to the developers.

Disappointingly, we encountered the XOR MAC algorithm late in this project and have not been able to experiment with it. For details of the algorithm, see [5]. (Philip Rogaway, one of the developers of XOR MAC, has developed a more efficient version, named PMAC [67]. As this report goes to press, the proof of PMAC's security has not been completed, so it must be considered "provisional.")

An additional approach that we have not yet discounted, is described by Krawczyk [38]. This method involves matrix multiplication of a Toeplitz NxM Boolean matrix with the message, which is considered to be an Mx1 matrix, to arrive at an Nx1 authentication tag. The tag depends on each bit of the message and the algorithm trivially lends itself to be run in N parallel streams. Note, however, that the running time of this algorithm is linear, whereas XOR MAC is logarithmic.

3.3.4 Advanced Encryption Standard Considerations

Rijndael has been selected by NIST as the algorithm for the Advanced Encryption Standard (AES). Rijndael was one of the finalist candidate algorithms that does appear to lend itself to high-speed implementations in ASICs or Programmable Logic Devices.

For AES implementation into an ASIC, the first hurdle is the number of necessary I/O signals. AES has been defined for a block length of 128 bits and key lengths of 128, 192, or 256 bits. Large numbers of signals such as this require a large bond-pad ring that may forcibly enlarge the minimum area of any ASIC. ASIC areas should be minimized to increase fabrication yields. To combat this it may be necessary to use fewer, high-speed I/O buffers rather than more voltage-level style traditional buffers. Another option, because of the large number of I/O pins, is to use two rings of I/O bondpads to keep the circumference dictated by the bondpads in close proportion to the area required by the algorithm logic.

4 Considerations for 80-160 Gbps

4.1 UTOPIA 6

With work to develop an OC-768 (UL5, 40 Gbps) specification only now beginning in the international standards bodies, it is pure speculation as to what the requirements, specifications, and time frame will be for OC-3072 (UL6, 160 Gbps). However, that is the task at hand. Therefore, an attempt to visualize the UL6 interface will be made based on the past history of high speed interface specifications developed in the ATM Forum (ATMF) and the Optical Internetworking Forum (OIF). Here this specification will be referred to as UTOPIA Level 6 (UL6), following on the previous UTOPIA series of specifications developed by the ATMF. These specifications are currently being developed in parallel at the OIF where they are referred to as SPI-4, SPI-5, SPI-6, etc.

These interfaces are used to specify the data transport mechanism between framers and serializer/deserializers (SERDES) for the physical layer device (fiber optics). Figure 15 shows a system level prospective and indicates where these interfaces occur.

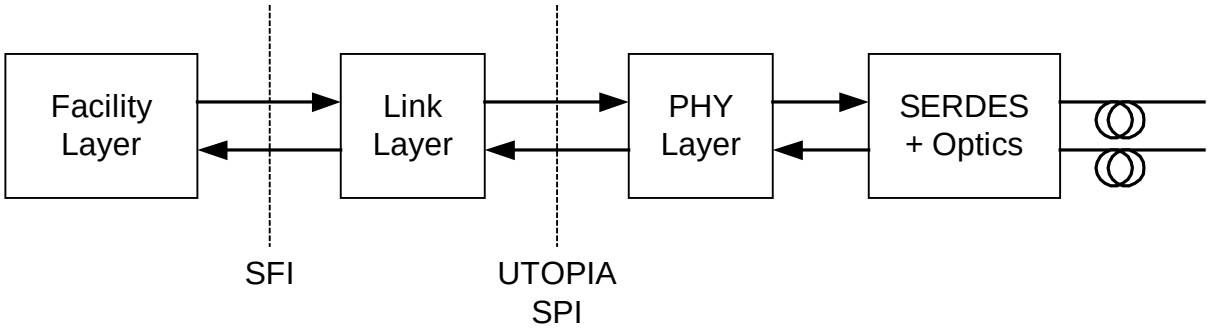


Figure 15. Interface layer definitions.

4.1.1 Standards Development History

At this point a little bit of history is in order. The ATMF has developed a series of chip-to-chip interface specifications called UTOPIA (Universal Test and Operations PHY Interface for ATM) that define the interface between the physical layer (PHY) and the upper layers of the ATM protocol (often referred to as the link layer). The OIF has been developing a similar set of specifications that are referred to as SPI (System Packet Interface). With the development of SPI-5 and SFI-5 (SERDES Framer Interface) the electrical interfaces appear to be converging on a single common interface. This interface will most likely be specified in a separate document published by the OIF that describes the physical and electrical details of the interface. These interface relationships are shown in Figure 15. While the PHY and SERDES have been drawn as separate, they can be

located in a common chip or module. The LINK layer usually contains the Sonet/SDH Framer or ATM Framer.

The publication date and data rate of the ATMF UTOPIA Standards and the OIF SPI standards is shown in Figure 16. As can be seen from past history, each new generation of optical network standards has increased the data rate by a factor of four over the previous standard. It should be pointed out that this technology step is different than that of the data networking (Ethernet) standards that have been increasing by factors of 10 over the past 20 years. The industry driven demand for higher bandwidth, wide area telecommunications and networking equipment has driven the throughput requirements up at a rate of one new generation of equipment every two years. It is hard to predict exactly when demand for a new generation of equipment will rise to the point that a new specification is needed. For that matter, the standards setting process is highly political with its own set of internal forces that lead to strange perturbations in the standard technology development curve. One such point can be seen in the apparent delay of the UL3 specification. Work was under way on a UL3 specification in September of 1997, however, a final specification was not passed until November of 1998. Development of the UL4 specification was well underway by the time UL3 was finalized and approved as a specification. Industry demand for UL4 (and subsequently FB4 and SPI-4) throughputs put the specification developments back on the technology track.

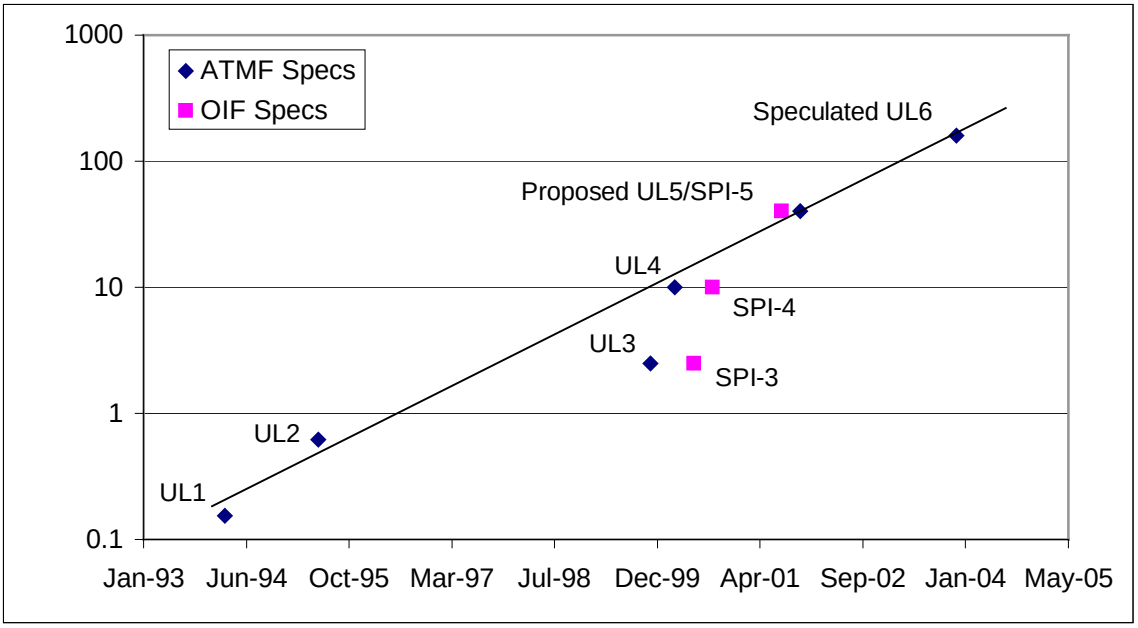


Figure 16. Specification approval dates for the past decade.

It should also be pointed out that there are several specifications that were not included in Figure 16. The ATMF published a Frame Based Extension to UL3, two Frame Based ATM specifications (one for Ethernet and one for Sonet/SDH) and a Frame Based ATM Level 4 specification that is nearly identical to the SPI-4 specification.

Using past history as our guide, it can be speculated that UL6 (OC-3072, 160 Gbps) will be needed in the December 2003 time frame. This seems fantastic given that only one year ago the industry was sorting out competing 10 Gbps specifications (UL4 versus SPI-4). There are certainly a number of hurdles to be crossed before we have silicon interfaces that operate at these rates. However, it seems inevitable that 160 Gbps will be a reality in the near future.

Table 3 contains a comparison of the basic characteristics of each of the interface specifications. As of January 2001, OC-192 (10 Gbps) specifications have been approved in both the OIF and the ATMF. Many in the industry have expressed their opinion that the next step up in speed might not even be 160 Gbps, but might be limited to 100 Gbps or 120 Gbps. However, past experience has shown that the demand for ever higher interface performance will not go away. Analysis of current silicon technology seems to indicate that there are no insurmountable barriers to achieving a 160 Gbps interface.

Table 3. Specification Comparison Chart.

ATMF Specs	Specification	Date	Data Rate (Gbps)	Nominal Bus Width	Physical Bus Width	I/O Specification	Nominal Line Rate MHz	Sonet
UL1	af-phy-0017.000	Mar-94	0.155	8	8	TTL	25	OC-1
UL2	af-phy-0039.000	Jun-95	0.622	16	16	LVTTL	50	OC-3
UL3	af-phy-0136.000	Nov-99	2.5	32	32	LVTTL	104	OC-48
UL4	af-phy-0144.001	Mar-00	10	32	64 (diff)	LVDS	315	OC-192
UL5	-	Nov-01	40	16	32 (diff)	LVDS	2500	OC-768
UL6	-	Dec-03	160	32	33	TBD	5000	OC-3072
FB3	99-0538	Feb-00	2.5	32	32		104	OC-48
FB4	fb-phy-0161.00	Nov-00	10	16	32 (diff)	LVDS	622 (311 double edge clocking)	OC-192

OIF Specs	Specification	Date	Data Rate (Gbps)	Nominal Bus Width	Physical Bus Width	I/O Specification	Nominal Line Rate MHz	Sonet
SPI-3	oif-pll-01-00	Jun-00	2.5	32	32	LVTTL	104	OC-48
SPI-4	oif-pll-03-00	Sep-00	10	16	32 (diff)	LVDS	622 (311 double edge clocking)	OC-192
SPI-5	oif-pll-04-00	Aug-01	40	16	32 (diff)	LVDS	2500	OC-768

4.1.2 Requirements Definition

In an effort to determine what UL6 will look like, a set of requirements can be developed given several considerations such as interface data rate, data path width, line speed, package and board layout considerations and power requirements.

Cost of implementing the interface as specified is also a consideration, but one that is less tangible, harder to quantify and involves a number of independent factors. If the interface is too wide, the package size must be increased. If the speed of a data line is specified too fast, exotic technologies such as GaAs or SiGe (i.e. not CMOS) must be tapped to

implement the interface and low loss dielectric materials must be used to fabricate the printed circuit boards. All these items will result in increased cost.

The speed of the interface will ultimately be determined by specification of the OC-3072 and STS-3072 data rates. Here it is assumed that the rate will be a 4x multiple of the OC-768 specification or 159.23 Gbps. There will be some additional overhead, normally associated with framing of the data. At OC-768 this overhead accounts for approximately 3 Gbps out of 40 Gbps or approximately 8 percent. An 8 percent overhead at OC-3072 would mean that the interface would have to operate at over 172 Gbps aggregate rate.

The number of lines used in the interface will be a trade off between having more lines thus lowering the individual line rate and supporting less lines and increasing the speed of the individual lines. Here we assume that the interface will consist of 16 logical lines operating at 10.75 Gbps each or 32 logical lines operating at 5.37 Gbps each. Given that the current 40 Gbps interfaces appear to be using 16 bits operating at 2.5 Gbps, it can safely be assumed that over the next three years, CMOS I/O technology will be capable of supporting over 5 Gbps on a single line (5 GHz technology). It is a bit harder to say that a single chip could support a bi-directional interface having 32 data lines (16 in each direction), each line consisting of differential signaling I/O interface such as LVDS or CML (Low Voltage Differential Signaling or Common Mode Logic). Differential signaling for 32 data lines requires 64 I/O pins. Therefore, a scheme must be utilized that will allow the interface to have 32 parallel bits, each operating as a single ended, independent data line. Single-ended signaling for 32 data lines requires only 32 pins. Support for deskew, forward error correction (FEC) and retiming must be independent and tolerant of a great deal of high frequency noise and jitter.

However, there are other problems with having an interface that requires such a large number of lines (even for 32 single-ended lines). Each line must be routed across a printed circuit board as a controlled impedance line (generally 50 ohms to ground or 100 ohms differential impedance in the case of differential logic family such as LVDS). State of the art ball grid packages have square arrays of balls on a 1mm (or less) grid with from 22 (484 total balls) to 26 (676 total balls) balls on a side. Consider the case of differential logic I/O. Differential lines must be bonded out to neighboring package pins. It is important that all lines be routed in parallel across the circuit card with the same lengths to insure high speed operation. Due to the large number of ground and power pins required on these packages, only a limited number of lines can be reasonably be used from a single edge of the package. As shown in Figure 17, a 676 ball package, can support a 16 bit, differential bus on each side of the package with routing limited to a single layer of the circuit board. A 32 bit bus would require a second row of balls on each side and those lines would have to be routed on at least two additional layers. An example of a 32 bit differential bus on the edge of a 676 pin BGA is shown in Figure 18. A device having both a bi-directional SFI interface and bi-directional SPI interface would have all four sides of its package effectively filled with high speed lines. While this configuration will not be easily realized, it is conceivable that such a device could be built. Obviously, getting optimum performance from such a device would be a signal integrity challenge.

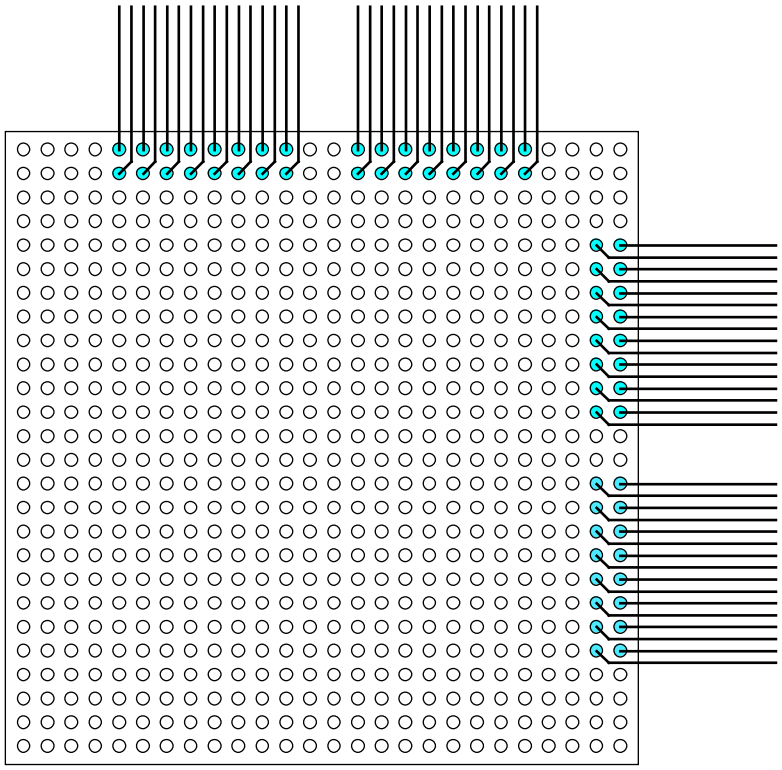


Figure 17. Example routing of SPI-6 interface with 32 single-ended data lines in a 676 pin BGA package.

It can therefore reasonably be assumed that the UL6 specification will rely on single ended interconnections of a width of 32 bits thus avoiding the problems associated with routing differential logic on the board and enabling the routing to be done on a single printed circuit board layer.

Current FR-4 circuit board has a limited bandwidth of around 3 Gbps or approximately 6 GHz for the clock, due to high signal loss at high frequencies. Recall that the clock requires approximately twice the bandwidth that the data does since the clock has two transitions per bit period. Circuit boards that will be used with the 160 Gbps generation of PHY devices will have to support bit rates of over 6 Gbps for each data line requiring approximately 12 GHz for the clock. Even though this is a simplified argument, it serves to demonstrate the desire and need to develop high speed interfaces that do not require sending the clock across the interface.

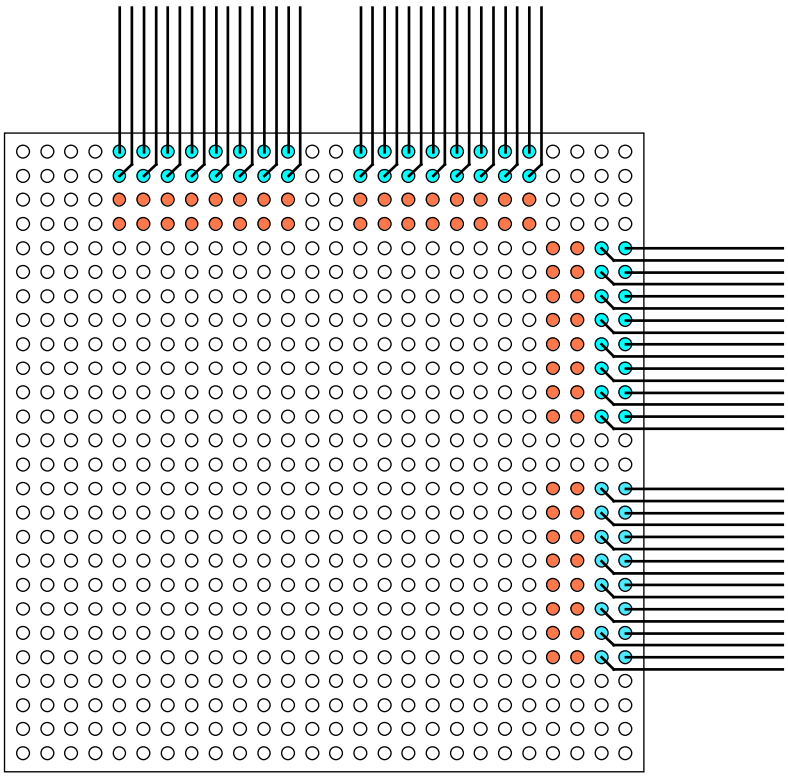


Figure 18. Routing of 32-bit differential interface in a 676 pin BGA package requiring additional lines and additional routing layers.

Special low loss dielectric materials will have to be used on the upper layers of the circuit board increasing the fabrication cost of the circuit board. It is possible that higher frequency circuit technologies such as Multi Chip Module (MCM) might be utilized, however, the cost of these technologies is significantly higher than the lowly copper printed circuit board technology. However, if it is possible to specify the interface so that the clock is not necessary, then the required signal bandwidth can easily be achieved using low loss dielectric materials and current printed circuit board technology. This might be done by encoding the clock in the data transitions of multiple scrambled data bits or by methods that distribute clock information over multiple lower frequency signals(e.g. quadrature clocking).

Other interface requirements will be similar to existing lower speed interfaces such as UL4 and UL5 or SPI-4 and SPI-5. Current development of the UL5/SPI-5 interface is shedding light on new requirements to support future high speed interfaces. Support for explicit signaling of FIFO status remains. It appears that out of band FIFO status will continue to be the norm. In the past history of the UTOPIA interface specifications, only the UL4 specification utilized in-band communications of the FIFO status information. It does not appear that the industry will support in-band FIFO status in the future. The interface must support communications over 4-8 inches of printed circuit board, most likely utilizing a low loss dielectric material. Support for a connector will only be possible with the introduction of very low loss controlled impedance connector designs.

It appears that these connectors are on the horizon and therefore it may be possible to support a connector.

Skew compensation must be supported. This must be achieved without transmitting a clock at twice the data rate across the interface as discussed previously. Each line must perform data boundary determination and forward error correction (FEC) independently of every other line. Deskew must be performed compatible with FEC and without regard to the framing scheme. Due to the tight timing constraints (the clock period of a 32 bit interface will be 200 ps), deskew must be continuously performed tracking changes in skew due to temperature and other environmental variations. Sending the clock (which will be running at 5 GHz) across the interface will require a channel bandwidth of at least 10 GHz (if not 18 GHz) to preserve edges and rise-times of the signal. It is therefore likely that the interface be defined in such a way that the clock is not necessary to the recovery of the data. The data interconnect bandwidth could then be limited to about half of that needed for the clock, or about 9 GHz. This bandwidth is achievable on printed circuit boards using low loss dielectrics under the high speed signal lines.

4.1.3 Summary

In summary, it is conceivable that the need for a 160 Gbps interface specification will push development of such a spec before the end of 2003. Speculatively called OC-3072, this interface might operate with an interface width of 16 differential data lines and at around 10.75 Gbps each. The interface will most likely not contain a clock line, but will support continuous deskew, perhaps utilizing additional data line(s). The interface will utilize out of band signaling of the FIFO status and support all other functions of interfaces such as UL4, FB4 and the upcoming UL5 and SPI-5. The device supporting this interface will most likely be fabricated from SiGe.

4.2 Encryption at 160 Gbps

For encryption of Asynchronous Transfer Mode (ATM) communication sessions (and possibly other communication technologies) where six encryption engines could operate in parallel on 64 bit blocks to encrypt a 384 bit payload, 40 Gbs (OC-768) rates could be achieved with six SNL DES ASICs. The authors would expect six parallel DES ASICs made using a GaAs process to support encryption at 160 Gbps and beyond. Also increasing the number of pipeline stages and moving to production processes with smaller, faster features, would increase encryption/decryption throughputs. It is expected that similar techniques could be applied to accelerate the AES Rijndael algorithm.

Key to the parallelism that will be required for encryption and decryption at rates in the 10 to 100 Gbps range, and beyond, is elimination of the Cipher Block Chaining (CBC) encryption mode. This mode is not able to be pipelined (for a single communication session) because of the feedback loop chaining one block of ciphertext to the next plaintext input block [23]. A better mode of operation for high speed encryption, where parallelism must be exploited, is Counter Mode [4]. This is discussed further in several references [87] [59].

5 Conclusions

This research effort has surveyed state-of-the-art techniques in several areas of network architecture, protocol processing, switching, routing, data transmission, and optical data propagation. It then applied itself to advancing that state-of-the-art and honing selected core competencies of SNL personnel working in these areas.

Certain promising methods for bypassing the overhead associated with communication processing within Operating Systems (OS Bypass) were studied, with the study results documented in Section 2.3 of this report. In addition, a simulation study was conducted on the VIA protocol, as described in [79]. The results of this study indicate that VIA provides low-latency transfers in large clusters – a result that is expected to apply to most OS bypass mechanisms in general.

Technology for optical transmission at different wavelength attenuation "windows" was examined, and architectural directions were studied. In spite of advances in optical amplification, long haul, Wide Area Networks will continue to be constructed with regenerators that reconstruct the digital signal and reset the signal-to-noise ratio. Streamlined data transmission protocols (such as the Simple Data Link protocol) will still require error measurement mechanisms to diagnose failed regenerators, separate from the error measurement mechanisms needed to assure reliable end-to-end communications. Ingenious all-optical means may evolve to accomplish this. Encryption will continue to be done in the electrical domain, mostly because so little research is being done to develop all-optical encryption. The boundary between optics and electronics in these communication systems will continue to shift more functionality into the optical domain. The lack of optical means for parallel-to-serial conversion currently limits optical processing to those operations that can be performed on a serial stream.

In the parallel electrical protocol processing world, the "tug-of-war" between wide words, wide busses, and large numbers of package pins (to keep clock rates low) and the ability to process faster and faster clock rates (enabling narrower busses and lower cost components with fewer package pins) will continue. Designers will begin to employ "source synchronous clocking" to make their designs more scalable to higher clock rates, automatic de-skewing of parallel data signals will become commonplace, and lower-loss dielectric materials will be used to enable higher controlled impedance traces on circuit boards. Protocols will eventually be re-designed to accomplish protocol processing "in systolic fashion", moving error checksums to the end of packets so that the entire packet need not be buffered during switching.

In the electrical domain, the current trend is to move data switching functions to higher layer protocols, i.e., to IP. By moving switching to the IP layer, the overhead associated with lower layer switching, headers, and duplication of network functions (e.g., path signaling and routing) is removed. However, integration of voice, video, and data will require true "quality of service" guarantees for different types of data, and will also

require means of auditing/billing reliably for consumption of communication resources required to deliver the appropriate qualities of services.

This work has facilitated the rapid maturation of high speed networking and communication technology by 1) participating in the development of pertinent standards, and 2) by promoting informal (and formal) collaboration with industrial developers of high speed communication equipment. Extensive contributions have been made to work items ongoing in the Security, Wireless, Physical Layer, and other working groups of the ATM Forum. Similarly, contributions have been made to the Optical Internetworking Forum on similar subjects. This work has put project members into close ongoing collaboration with others who are pioneering work in these areas from NSA, NRL, Lucent, General Dynamics, PMC-Sierra, Vitesse, Xilinx, Altera, and other government and industrial partners.

6 Bibliography

1. L. Andersson, et. al., LDP Specification, *Internet Engineering Task Force RFC 3036*, January, 2001.

2. "APA Optics, Inc. Reports Orders From Two Major Telecom Companies for Its New 16-Channel WDM Multiplexers," press release, August 3, 1998.

3. ATM Continues Growth in WAN and LAN, in *Business Communications Review*, vol. 29, pp. 8, October 1999.

4. ATM Security Specification Version 1.0, AF-SEC-0100.001, ATM Forum, Mountain View, CA, February 1999.

5. Mihir Bellare, Roch Guerin, and Phillip Rogaway, XOR MACs: New Methods for Message Authentication Using Finite Pseudorandom Functions, in *Crypto '95*, 1995.

6. V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, Kluwer Academic Publishers, Boston, MA, 1997.

7. Blumrich, et al., Virtual-Memory-Mapped Network Interfaces, in *IEEE Micro*, vol. 15, pp. 21-28, February 1995.

8. Boden, et al., Myrinet: A Gigabit-per-Second Local Area Network, in *IEEE Micro*, vol. 15, pp. 29-36, February 1995.

9. Paul Bonenfant, Antonio Rodriguez-Moral, and James Manchester, IP Over WDM: The Missing Link, in *NFOEC'99*, 1999.

10. A. Brodnik, S. Carlsson, M. Degermark, and S. Pink, Small Forwarding Tables for Fast Routing Lookups, in *Proceedings of ACM SIGCOMM '97*, September 1997.

11. C. Brown, Lucent Cracks the Terabit Network Barrier, in *EE Times*, January 28, 1998.

12. Robert K. Butler and David R. Polson, Wave-Division Multiplexing in the Sprint Long Distance Network, in *IEEE Communications Magazine*, vol. 36, pp. 52-55, February 1998.

13. R. Callon, Multi-Layer Switching, in *Tutorial Proceedings of ATM Year 98*, June 1998.

14. Phillip L. Campbell, *An Implementation of the Berlekamp-Massey Linear Feedback Shift-Register Synthesis Algorithm in the C Programming Language*, SAND99-2033. Sandia National Laboratories, Albuquerque, NM, August 1999.

15. Phillip L. Campbell and Lyndon G. Pierson, *LFSRs Do Not Provide Compression*, SAND99-2892. Sandia National Laboratories, Albuquerque, NM, December 1999.

16. Capturing Rainbows, *America's Network*, May 15, 1998, pp. 4-5.

17. Compaq, Intel, and Microsoft, *Virtual Interface Architecture Specification*, Draft Revision 1.0, December 4, 1997.

18. Robert Cubbage, Proposed Structure Change to Section 3 of 2488.32 Mbps Phy Spec, *ATM Forum Contribution 97-0142*, February 10, 1997.

19. Vittorio Curri, System Advantages of Raman Amplifiers, in *Technical Proceedings, 16th Annual National Fiber Optic Engineers Conference (NFOEC)*, held in Denver, CO, August 27-31, 2000. NFOEC, 2000.

20. Data Encryption Standard, FIPS PUB 46-2, National Bureau of Standards, Washington, D.C., 1993.

21. B. Davie, P. Doolan, and Y. Rekhter, *Switching in IP Networks: IP Switching, Tag Switching, & Related Technologies*, Morgan Kaufmann, 1998.

22. S. Deering, and R. Hinden, Internet Protocol, Version 6 (IPv6), RFC-1883, *Internet Request for Comments, 1883*, December 1995.

23. DES Modes of Operation, FIPS PUB 81, National Bureau of Standards, Washington, D.C., 1980.

24. Dunning, et al., The Virtual Interface Architecture, in *IEEE Micro*, vol. 18, pp. 66-76, March/April 1998.

25. Frame-based ATM Interface (Level 3), AF-PHY-0143.000, ATM Forum, Mountain View, CA, March 2000.

26. Jerrold A. Friesen, and Thomas D. Tarman, Remote High-Performance Visualization and Collaboration, in *IEEE Computer Graphics and Applications*, vol. 20, pp. 45-49, July/August 2000.

27. J. Gamelin and R. S. Vodhanel, Healthy Skepticism Greets WDM Claims, in *Telephony*, pp. 34-38, May 11, 1998.

28. R. Genin and J. M. Daza, Blinded By the Wave-Division Light, in *Network World*, pp. 45-46, June 15, 1998.

29. P. Gupta, S. Lin, and N. McKeown, Routing Lookups in Hardware at Memory Access Speeds, in *IEEE INFOCOM '98*, April 1998.

30. Gilbert Held, *Data Compression*, John Wiley & Sons, New York, 1983.

31. A. Huang, Challenges in the Design of a 100 Gb/s Internet, in *Proceedings of the Second International Conference on Massively Parallel Processing Using Optical Interconnections,* IEEE Computer Society Press, Los Alamitos, CA, 1995.

32. Inverse Multiplexing for ATM (IMA) Specification, Version 1.1, af-phy-0086.001, The ATM Forum, Mountain View, CA, March 1999.

33. Charanjit S. Jutla, Encryption Modes with Almost-Free Message Integrity, presented at the NIST AES Modes of Operation Workshop, held in Baltimore, MD, October 16, 2000 (http://csrc.nist.gov/encryption/modes/workshop1/papers/jutla-auth.pdf).

34. S. Keshav, and Rosen Sharma, Issues and Trends in Router Design, in *IEEE Communications Magazine*, vol. 36, pp. 144-151, May 1998.

35. Joseph R. Kiniry, Wavelength Division Multiplexing: Ultra High Speed Fiber Optics, in *IEEE Internet Computing*, vol. 2, pp. 13-15, March-April 1998.

36. Rob Koenen, MPEG-4: Multimedia for Our Time, in *IEEE Spectrum*, vol. 36, pp. 26-33, February 1999.

37. T. Krause, The Emerging Optical Layer, in *Telephony*, pp. 92-94, June 8, 1998.

38. Hugo Krawczyk, New Hash Functions for Message Authentication, in *Eurocrypt '95*, pp. 301-10, 1995.

39. Vijay P. Kumar, T. V. Lakshman, and Dimitrios Stiliadis, Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet, in *IEEE Communications Magazine*, vol. 36, pp. 152-164, May 1998.

40. P. Lagasse, et al., *Roadmap Towards the Optical Communication Age, A European View by the HORIZON Project and the ACTS Photonic Domain*, ACTS HORIZON Project, June 1998 (Draft).

41. C.J. Loberg, Testing the Future of All-Optical Networks, in *Lightwave*, March 1998.

42. J. L. Massey, Shift-Register Synthesis and BCH Decoding, in *IEEE Transactions on. Information Theory,* pp. 122, 1969.

43. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

44. Ralph C. Merkle and Martin E. Hellman, On the Security of Multiple Encryption, in *Communications of the ACM*, vol. 24, pp. 465-467, July 1981.

45. Ron Minnich, Dan Burns, and Frank Hady, The Memory-Integrated Network Interface, in *IEEE Micro*, vol. 15, pp. 11-20, February 1995.

46. National Committee for Information Technology Standardization (NCITS) Committee T11.1, *Information Technology – Scheduled Transfer Protocol*, Working Draft, March 1998.

47. Mark Nelson, and Jean-Loup Gailly, *The Data Compression Book*, Henry Holt and Co., New York, 1996.

48. P. Newman, G. Minshall, and T. Lyon, IP Switching: ATM Under IP, in *IEEE/ACM Transactions on Networking*, vol. 6, pp. 117-129, April 1998.

49. P. Newman, G. Minshall, T. Lyon, and L. Huston, IP Switching and Gigabit Routers, in *IEEE Communications Magazine*, vol. 35, pp. 64-69, January 1997.

50. S. Nilsson, and G. Karlsson, Fast Address Lookup for Internet Routers, in *Fourth IFIP Conference on Broadband Communications*, April 1998.

51. S. Nilsson, and G. Karlsson, Fast IP Routing with LC-Tries, in *Dr. Dobb's Journal*, August 1998.

52. Tomohiro Otani, Tetsuya Miyazaki, and Shu Yamamoto, 40 Gbit/s Optical 3R Regenerator Using Electroabsorption Modulators for High-Speed Optical Networks, in *Technical Proceedings, 16th Annual National Fiber Optic Engineers Conference (NFOEC)*, held in Denver, CO, August 27-31, 2000. NFOEC, 2000.

53. Scott Pakin, Vijay Karamcheti, and Andrew A. Chien, Fast Messages: Efficient, Portable Communication for Workstation Clusters and MPPs, in *IEEE Concurrency*, vol. 5, pp. 60-73, April-June 1997.

54. Craig Partridge, *Gigabit Networking*, Addison Wesley, Reading, MA, 1994.

55. Craig Partridge, Using the Flow Label Field in IPv6, *Internet Request for Comments, 1809*, June 1995.

56. Craig Partridge, et al. "A 50-Gb/s IP Router," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 237-248, June 1998.

57. G. Parulkar, D. C. Schmidt, and J. Turner, IP/ATM: A Strategy for Integrating IP with ATM, in *Proceedings of ACM SIGCOMM*, October 1995.

58. L. G. Pierson and J. H. Maestas, *Efficient Synchronization of Bit-Error Communication Test Equipment*, SD-5994, April 1997.

59. Lyndon G. Pierson, Thomas, D. Tarman, and Edward L. Witzke, *Scalable End-To-End Encryption Technology for Supra-Gigabit/second Networking*, SAND94-1622. Sandia National Laboratories, Albuquerque, NM, May 1997.

60. Lyndon G. Pierson and Edward L. Witzke, *Parallel Generation of Linear Recurring Sequences*, SD-6013, May 1997.

61. Lyndon G. Pierson, Edward L. Witzke, Mark O Bean, and Gerry J Trombley, Context-Agile Encryption for High Speed Communication Networks, in *ACM Computer Communication Review*, vol. 29, pp. 35-49, 1999.

62. Martin de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood Ltd., New York, 1993.

63. Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, Cisco Systems' Tag Switching Architecture Overview, *Internet Request for Comments, 2105*, February 1997.

64. Perry J. Robertson, Robert L. Hutchinson, Lyndon G. Pierson, Thomas D. Tarman, and Edward L. Witzke, *Final Report and Documentation for the PLD11 Multipurpose Programmable Logic VME Board Design*, SAND99-0914. Sandia National Laboratories, Albuquerque, NM, April 1999.

65. Perry J. Robertson and Edward L. Witzke, *Multiply-Fed CBC Mode Encryption*, SD-6769, October 2000.

66. Perry J. Robertson and Edward L. Witzke, *Aggregate Encryptor/Scattering Decryptor*, SD-6770, October 2000.

67. Phillip Rogaway, PMAC: A Parallelizable Message Authentication Code, presented at the NIST AES Modes of Operation Workshop, held in Baltimore, MD, October 16, 2000 (http://csrc.nist.gov/encryption/modes/workshop1/papers/rogaway-pmac1.pdf).

68. E. Rosen, A. Viswanathan, and R. Callon, Multiprotocol Label Switching Architecture, *Internet Engineering Task Force RFC 3031*, January, 2001.

69. E. Rosen, et. al., MPLS Label Stack Encoding, *Internet Engineering Task Force RFC 3032*, January, 2001.

70. John P. Ryan, WDM: North American Deployment Trends, in *IEEE Communications Magazine*, vol. 36, pp. 40-44, February 1998.

71. Bruce Schneier, *Applied Cryptography*, 2nd ed., John Wiley & Sons, New York, 1996.

72. R. A. Shaffer, The Next Big Switch Will Be Optical, in *Fortune*, pp. 150-153, June 22, 1998.

73. Peter Sholander, Thomas Tarman, Lyndon Pierson, and Robert Hutchinson, The Effect of Algorithm-Agile Encryption on ATM Quality of Service, in *Proceedings, Globecom '97*, held in Phoenix, AZ, November 1997. IEEE Computer Society Press, Los Alamitos, CA, 1997.

74. Gustavus J. Simmons, ed., *Contemporary Cryptology*, IEEE Press, Piscataway, NJ, 1992.

75. Mark J. Soulliere, Proposed Optical Parameters for 2488.32 Mbps Single Mode Fiber Interface, *ATM Forum Contribution 97-0127*, February 10, 1997.

76. "Sprint Increases Network Capacity Performance With Deployment of CIENA's Scaleable 40-Channel MultiWave 4000," press release, March 16, 1998.

77. V. Srinivasan, and G. Varghese, Faster IP Lookups using Controlled Prefix Expansion, in *SIGMETRICS '98/PERFORMANCE '98*, June 1998.

78. Thomas D. Tarman, Robnert L. Hutchinson, Lyndon G. Pierson, Peter E. Sholander, and Edward L. Witzke, Algorithm-Agile Encryption in ATM Networks, in *IEEE Computer*, vol. 31, no. 9, pp. 57-64, 1998.

79. Thomas D. Tarman, Leonard Stans, and Tan Chang Hu, A Simulation Study of the Virtual Interface Architecture, in *Proceedings, OPNETWORK '99*, held in Washington D.C., August, 1999.

80. S. Thomas, *IPng and the TCP/IP Protocols*, John Wiley & Sons, New York, 1996.

81. J. Touch, Report on MD5 Performance, *Internet Request for Comments, 1810*, June, 1995.

82. Triple Data Encryption Algorithm Modes of Operation, (ANSI) ANS X9.52, American Bankers Association, Washington, D.C., 1998.

83. Gerry J. Trombley and Mark O. Bean, *Technology Trends Influencing High-Speed INFOSEC Requirements*, R2 Technical Report R22-003-98. National Security Agency, Ft. Meade, MD, February 1998.

84. Unleashing the Power of Light, *Network Edge*, Winter 1998-1999, pp. 3.

85. UTOPIA Level 4, AF-PHY-0144.001, ATM Forum, Mountain View, CA, March 2000.

86. M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, Scalable High Speed IP Routing Lookups, in *Proceedings of ACM SIGCOMM '97*, September 1997.

87. D. Craig Wilcox, Lyndon G. Pierson, Perry J. Robertson, Edward L. Witzke, and Karl Gass, A DES ASIC Suitable for Network Encryption at 10 Gbps and Beyond, in *Cryptographic Hardware and Embedded Systems*, Vol. 1717 of *Lecture Notes in Computer Science*, held in Worcester, MA, August 12-13, 1999. Springer-Verlag, Berlin, 1999.

88. Kwang-Jin Yang, et al., Optical Transponder with Bit-Rate Independent Clock and Data Recovery (BICDR) for DWDM Systems, in *Technical Proceedings, 15th Annual National Fiber Optic Engineers Conference (NFOEC)*, held in Chicago, IL, September 26-30, 1999. NFOEC, 1999.

Appendix, LDRD Data

This effort was funded by Sandia National Laboratories' Laboratory Directed Research and Development program under cases 3512250000 and 3512440000, and project 10367.

Awards: N/A

Publications:
Thomas D. Tarman, Leonard Stans, and Tan Chang Hu, A Simulation Study of the Virtual Interface Architecture, in *Proceedings, OPNETWORK '99,* 1999.

D. Craig Wilcox, Lyndon G. Pierson, Perry J. Robertson, Edward L. Witzke, and Karl Gass, A DES ASIC Suitable for Network Encryption at 10 Gbps and Beyond, in *Cryptographic Hardware and Embedded Systems*, Vol. 1717 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1999.

Lyndon G. Pierson, Edward L. Witzke, Mark O. Bean, and Gerry J. Trombley, Context-Agile Encryption for High Speed Communications Networks, in *ACM Computer Communication Review*, Vol. 29, no. 1, pp. 35-49, 1999.

During this LDRD, 15 contributions to the ATM Forum Technical Committee were prepared/submitted/presented.

Patents (applied for or issued):
A provisional patent has been filed by Sandia covering Technical Advances SD-6769 and SD-6770.

Technical Advances filed under this LDRD include:
Perry J. Robertson and Edward L. Witzke, *Aggregate Encryptor/Scattering Decryptor*, SD-6770, October 2000.

Perry J. Robertson and Edward L. Witzke, *Multiply-Fed CBC Mode Encryption*, SD-6769, October 2000.

Lyndon G. Pierson, Perry J. Robertson, and Edward L. Witzke, *'ATM Bus' Based Computer,* SD-6265, August 1998.

L. Byron Dean II, *Combinator "Anycast" MMP Communications/computations,* SD-6193, May 1998.

Copyrights (for software): N/A

Employee recruitment: This project provided the initial funds to support a contractor conversion new hire (Edward Witzke).

Student Involvement: This project supported a summer intern (Mayfann Ngujo) from New Mexico State University.

Follow-on work (new activities, projects, sponsors): This project and our collaborations under it, facilitated a Work-For-Others task from NSA to examine issues with high-speed IP encryption

DISTRIBUTION:

1 Xilinx Inc.
Attn: D. Craig Wilcox
7801 Jefferson NE
Albuquerque, NM 87109

1	MS 0188	C. E. Meyers, 1030	1	MS 1072	K. K. Ma, 1735
1	MS 0188	LDRD Office, 1030	1	MS 1138	B. N. Malm, 6531
1	MS 0449	V. A. Hamilton, 6514	1	MS 1155	J. G. Peña, 6532
1	MS 0449	R. L. Hutchinson, 6516	1	MS 9003	P. W. Dean, 8903
5	MS 0449	P. L. Campbell, 6516	1	MS 9011	B. V. Hess, 8910
1	MS 0449	B. P. Van Leeuwen, 6516	1	MS 9011	H. Y. Chen, 8910
1	MS 0451	R. E. Trellue, 6501			
1	MS 0455	R. S. Tamashiro, 6517			
1	MS 0503	A. L. Schauer, 2341			
1	MS 0801	M. O. Vahle, 9300			
1	MS 0806	L. Stans, 9336			
1	MS 0806	J. P. Brenkosh, 9336			
5	MS 0806	L. B. Dean, 9336			
1	MS 0806	J. M. Eldridge, 9336			
1	MS 0806	M. J. Ernest, 9336			
1	MS 0806	S. A. Gossage, 9336			
1	MS 0806	R. L. Hartley, 9336			
1	MS 0806	T. C. Hu, 9336			
1	MS 0806	J. A. Hudson, 9336			
1	MS 0806	B. R. Kellogg, 9336			
1	MS 0806	L. G. Martinez, 9336			
1	MS 0806	M. M. Miller, 9336			
1	MS 0806	J. H. Naegle, 9336			
10	MS 0806	L. G. Pierson, 9336			
1	MS 0806	T. J. Pratt, 9336			
1	MS 0806	J. A. Schutt, 9336			
5	MS 0806	T. D. Tarman, 9336			
1	MS 0806	L. F. Tolendino, 9336			
10	MS 0806	E. L. Witzke, 9336			
1	MS 0806	R. R, Olsberg, 6531			
1	MS 0812	M. R. Sjulin, 9330			
1	MS 0812	J. H. Maestas, 9334			
1	MS 0874	D. W. Palmer, 1751			
5	MS 0874	P. J. Robertson, 1751			
1	MS 0874	K.L. Gass, 1751			

1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0612	Review & Approval Desk, 9612 For DOE/OSTI